Convex Relaxations in Mixed Integer Optimization Methods and Control Applications

Author(s): Vujanic, Robin

Publication Date: 2014

Permanent Link: https://doi.org/10.3929/ethz-a-010280590

Rights / License: In Copyright - Non-Commercial Use Permitted

This page was generated automatically upon download from the ETH Zurich Research Collection. For more information please consult the Terms of use.
Convex Relaxations in Mixed-Integer Optimization

Methods and Control Applications

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by

Robin Vujanic
M. sc. in Mechanical Engineering, ETH Zurich
born 31.07.1984
citizen of Switzerland

accepted on the recommendation of

Prof. Dr. Manfred Morari (examiner)
Prof. Dr. Ignacio E. Grossmann (co-examiner)

2014
“It is the struggle itself that is most important. We must strive to be more than we are. It does not matter that we will never reach our ultimate goal. The effort yields its own rewards.”

– Lieutenant Commander Data
Acknowledgements

This was a long journey, and it was made possible mainly by the people I have shared time with throughout those years. I wish there was a way to express gratitude towards all these people without sounding cheesy.

I would like to start from Prof. Manfred Morari, who welcomed me in his lab and gave me the opportunity to deepen my knowledge in the fields of automation and optimization. The time spent at IfA had a significant impact on me not only academically, but also on a personal level. And much of this is due to the discussions I had with him. I would also like to thank Prof. Ignacio Grossmann from Carnegie Mellon University, for refereeing my thesis but even more importantly for exposing me to the field of mixed integer optimization; a good part of my work grew from ideas I started to cultivate during one of his courses several years ago.

I am also thankful to my work colleagues. Peyman Mohajerin Esfahani, whom it was a great pleasure working with, I wish we started our collaborations earlier. The nights we spent together in the office trying to untangle mathematical messes will not be forgotten. You have been a fantastic office mate, your moral support throughout these years cannot be overstated, and your contributions to our works have been invaluable. I also want to thank Robert Nguyen, Martin Herceg and Alex Fuchs. Robert never failed to amaze me with his engineering pieces of art, ranging from automated plant watering systems up to the control of temperature in stoves for optimized cooking. And Martin and Alex, I had a fantastic time with you during our frequent lunches together. You were great work colleagues but even better friends. I also have to be deeply thankful to Paul Goulart and Sébastien Mariéthoz, my two supervisors, who were crucial for the success of my work. In particular, I am grateful for Séba’s support, especially in the most difficult periods of my PhD, and for helping me direct my research efforts towards sensible applications in control. And Paul I have to thank for his help and support on the theoretical problems we faced; he was a main contributor in pushing the quality of my work to the highest standards. And Stefan Richter, who exposed me to some of the papers that most influenced my works and the discussions with whom were always highly instructive. And finally Aldo Zgraggen, Christian Conte, Claudia Fischer, Joe Warrington, Marko Tanaskovic and Stefan Huck (in alphabetic order!), the group of PhD students who started roughly at the same time as me: we all moved to the next station or are
about to, but it was a great pleasure being work colleagues and spending time together in and outside of the lab. At this stage I also would like to thank Debasish Chatterjee, my old office mate, and Federico Ramponi: the impact they had on me while at IfA was greater than they may think.

But support came also outside of the work place. First of all I would like to thank my mother, for supporting my decisions so far, her boundless care for me, and for tolerating me as a son. Among my closest friends I would also like to express my gratitude to Matteo Bonelli, for all the time we spend together and for being a great friend, you made me smile during the thoughest times; Juerg Granwehr, for the company and the deep discussions about philopphical matters and life (though I never hoped to win an argument against a lawyer!); and then Yorgos Almpanis, Andre Bertolace, Andrea Giovannini and Marco Righettoni, my long time friends, you will always have a special place in my heart.

Thanks to all of you once again,
Robin.
Abstract

Systems that involve discrete components (e.g., on/off decisions, switches and symbolic reasoning) and continuous quantities (such as physical measurements of voltages, concentrations and positions in space) arise in several practical and industrial contexts. However, when decision problems associated to these systems are cast as mathematical optimization problems, the resulting models are affected by severe computational challenges. Even modest problem sizes can be unsolvable by general purpose solvers when the models contain discrete variables. Further, the required computation time and memory can vary significantly between two instances of the same size, which hinders the practitioners’ ability in allocating sufficient computational resources when such optimization problems have to be solved repeatedly – as is typically the case in a control context. These impediments significantly inhibit the usefulness of discrete models in practical applications.

To alleviate these difficulties, in this Thesis we have developed computational schemes to obtain approximate solutions for several problem structures that are of wide practical interest. They are computationally attractive, and equipped with guarantees on the approximation strength.

Two main approaches are used in order to derive our schemes.

The Lagrangian duality framework is the first, and it is used to tackle large scale, structured optimization problems. The ever increasing availability of large amounts of data and communication (the “big data” revolution), as well as the appeal in operating more efficiently large engineered systems, such as power systems, are some of the reasons why large scale optimization is flourishing. The Lagrangian duality framework is useful in this context because it allows one to decompose the problem in smaller pieces, but in contrast to the convex counterpart, when applied to models with discrete decisions (mixed-integer optimization problems) it is generally unable to produce global optimal solutions. In fact, the solutions recovered may not only be suboptimal, but even infeasible. In this Thesis we derive new structural properties of the solutions that can be recovered from the dual, and using these results we develop solution methods that are guaranteed to produce globally feasible solutions in a distributed fashion. They are simple to implement and the quality of the solutions they produce improves (in relative terms) as the size of the problems is increased. Roughly speaking, the reason why we can achieve this is that these instances
resemble more and more convex problems the larger they become. We illustrate the efficacy of our methods with extensive experimentations on problems stemming from power systems management and supply chain optimization.

The second approach is based on linear relaxations, and it is used in the context of optimizing systems affected by uncertainty. We investigate two structures in this setting.

In the first one, we look into problems affected by endogenous uncertainty, i.e., by uncertainty that depends on the decisions taken. We derive a robust counterpart in this case, and show how this framework can be useful for scheduling problems. As an illustrative example, we study the problem of integrating industrial power consumers in network reserve mechanisms.

The second one is related to the robust control of mixed-integer input linear systems. We derive a control scheme in which the mixed-integer optimal control problem is first relaxed, and a projection function is applied to the resulting relaxed control sequence. This procedure rectifies it into a solution satisfying the integrality conditions on the inputs, while state constraints satisfaction is guaranteed by applying a robust reformulation of the problem. The key ingredient here is the design of a suitable projection function. We find the class of Pulse-Width-Modulated (PWM) systems particularly well suited for this approach: we provide a projection function for this case and demonstrate how our controller performs on different converter topologies. The results indicate a substantial performance improvement with respect to a plain MPC implementation – at the same computational cost. We finally investigate approaches to reduce the necessary conservatism by adapting the idea of affine recourse for the robust control of MLD hybrid systems.
Compendio

Sistemi che contengono sia componenti discrete (come ad esempio decisioni di accensione o spegnimento, interruttori e ragionamenti logici) come anche quantità continue (ad esempio, misure di voltaggi, concentrazioni o posizioni nello spazio) appaiono in molti contesti pratici ed industriali. Tuttavia, quando i problemi decisionali associati a questi sistemi vengono espressi come problemi di ottimizzazione matematica, i modelli risultanti sono soggetti a notevoli difficoltà computazionali. Tant’è vero che persino istanze di moderate dimensioni possono essere irrisolvibili da software di soluzione generici. Inoltre, il tempo e la memoria richiesti per calcolare una soluzione possono variare in modo significativo tra due istanze della stessa dimensione. Ciò ostacola il professionista nello stabilire risorse computazionali sufficienti, soprattutto quando questi problemi di ottimizzazione devono essere risolti più e più volte – come è spesso il caso in un contesto di automazione. Questi impedimenti inibiscono in modo considerevole l’utilità di modelli discreti per applicazioni pratiche.

Per alleviare queste difficoltà, in questa Tesi abbiamo sviluppato degli schemi computazionali che permettono di calcolare soluzioni approssimative per varie strutture di problemi di interesse pratico. I nostri schemi hanno proprietà computazionali attrattive e sono equipaggiati di garanzie sulla qualità delle approssimazioni.

Abbiamo impiegato due approcci principali nella derivazione dei nostri metodi di approssimazione.

Il framework di dualità lagrangiana è il primo, ed è utilizzato per affrontare problemi di ottimizzazione di grosse dimensioni. La grande quantità di dati che diventano sempre più disponibili (la rivoluzione “big data”), come anche l’attrattività di operare in modo più efficiente grossi sistemi ingegneristici, come ad esempio sistemi di potenza concernenti intere reti elettriche, sono alcune tra le ragioni per cui l’ottimizzazione di problemi di grandi dimensioni sta acquistando importanza. Il framework di dualità lagrangiana è utile in questo contesto perché permette la scomposizione dei problemi in pezzi più piccoli ma, a differenza del caso convesso, non è generalmente in grado di fornire soluzioni ottimali globali quando applicato a modelli contenenti decisioni discrete. Anzi, le soluzioni ricavate possono essere non solo subottimali, ma persino violare i vincoli del problema. In questa Tesi abbiamo derivato nuove proprietà strutturali delle soluzioni che possono essere ottenute dal problema duale, e utilizzando questi risultati abbiamo sviluppato metodi che
producono soluzioni che soddisfano i vincoli sia locali che globali in modo distribuito. Sono semplici da implementare, e più le dimensioni dei problemi considerati sono grandi, migliore è la qualità delle soluzioni che producono (in senso relativo). Il motivo per cui è possibile ottenere questi risultati è che più grandi sono le istanze considerate, più queste assomigliano a problemi convessi. L'efficacia dei nostri metodi è estensivamente illustrata con sperimentazioni su problemi provenienti dall'operazione ottimale di sistemi di potenza e dall'ottimizzazione di supply chains.

Il secondo approccio è basato su rilassamenti lineari, ed è utilizzato nel contesto di ottimizzazione di sistemi influenzati da componenti incerte. Abbiamo investigato due strutture in questo senso.


La seconda è associata al controllo robusto di sistemi lineari con entrate continue e discrete. Abbiamo sviluppato uno schema in cui il problema di controllo ottimale viene dapprima rilassato, e la sequenza di inputs prodotta viene poi passata a una funzione di proiezione. Questa funzione rettifica gli inputs di modo che soddisfino le condizioni di integralità richieste, mentre i vincoli di stato vengono assicurati attraverso una riformulazione robusta del problema. La componente chiave del nostro sistema è la funzione di proiezione utilizzata. Troviamo che la classe di sistemi modulati a pulsi (PWM) sia particolarmente adatta a questo approccio: forniamo una funzione di proiezione adatta per questi sistemi e mostriamo la performance del controllore su diverse topologie di convertitori di potenza. I nostri risultati indicano che è possibile ottenere miglioramenti di performance sostanziali rispetto a un'implementazione di controllo predittivo comune – allo stesso costo computazionale. Per concludere abbiamo investigato metodi per ridurre il conservatismo necessario, adattando idee di ricorso affine per il controllo di sistemi ibridi.
1. Introduction

Optimization problems entailing discrete decisions arise in several control and automation contexts. Finding optimal solutions to these models is generally computationally hard; indeed, exact solutions can elude general purpose solvers even when the models involved are of modest size. Modern solvers currently operate using a combination of branch and bound and cutting plane methods (together with preprocessing and runtime heuristics). There is generally no way of telling, a priori, how much memory these methods will require to converge -- the cutting plane procedure in particular can be very memory-intensive. Further aggravating this issue is that computational requirements (CPU, memory and time) not only depend on the problem structure and size, but often also on the problem data. Hence, a given computational budget may be sufficient to solve a certain instance, but not necessarily the next one. On top of all this, the best solvers currently available are expensive proprietary software tools. These difficulties are a tremendous obstacle to the usefulness of discrete optimization models in broad practice.

In this thesis we investigate several particular model structures that are of practical interest. For these, we derive convex relaxations and provide guarantees on their tightness. These allow the design of computationally attractive approximation schemes.

Two main approaches are studied to obtain the relaxation schemes, and they determine the two parts of the Thesis.

Part I. The first approach is based on Lagrangian duality, and is used in the context of large scale optimization. The material presented in this part stems from [Vujanic et al., 2013a, Vujanic et al., 2014b, Vujanic et al., 2014a].

Lagrangian duality in mixed integer optimization is a useful framework for producing tight lower bounds to the optimal objective, but in contrast to the convex counterpart, it is generally unable to produce optimal solutions directly. In fact, solutions recovered from the dual may be not only suboptimal, but even infeasible. In this part of the thesis we concentrate on large-scale mixed integer programs with a specific structure that is of widespread practical interest, as it appears in a variety of application domains such as power systems or supply chain management.

Based on geometric properties of the primal solutions recovered, we can quantify the tightness of the relaxation achieved through the dual. Using this result, we propose a
solution method in which the primal problem is modified in a certain way, guarantee-
ning that the solutions produced by the corresponding dual are feasible for the original
unmodified primal problem. The modification accounts for the discrepancy introduced
by the relaxation. It is simple to implement and the method is amenable to distributed
computations. We also demonstrate that the quality of the solutions recovered using our
procedure improves as the problem size increases, making it particularly useful for large
scale instances for which commercial solvers are inadequate. Borrowing ideas and results
from the convex optimization literature, we then extend this approach by providing a
method endowed with convergence guarantees.

We illustrate the efficacy of our methods with extensive experimentations on problems
stemming from power systems management and supply chain optimization.

Part II. The second approach is based on linear relaxations of the problem, in the context
of robust optimization of uncertain systems involving discrete variables. The material
presented in this part stems from [Vujanic et al., 2012, Vujanic et al., 2013b, Schmitt
et al., 2013].

In this setting, we have investigated two types of structures.

The first is related to problems affected by endogenous uncertainty, i.e., by uncertainty
that depends on the decisions taken. We derive a robust counterpart in this case, and
show how this framework can be useful for scheduling problems. The uncertainty investi-
gated has the structure of a multiple choice problem, and its convex relaxation is used to
derive an explicit robust counterpart whose computational complexity is identical to the
deterministic case (i.e., computing a schedule neglecting uncertainties). As a practical
example, we show how the problem statement and the proposed robustification proce-
dure can be useful in the context of scheduling the consumption of industrial electricity
consumers, and their participation in network reserves mechanisms.

The second structure studied pertains model predictive control of dynamic systems with
mixed discrete and continuous inputs subject to disturbances. We discuss an approxi-
mate approach in which the integer input constraints are initially relaxed. A projection
is then applied to the relaxed solution in order to obtain inputs satisfying the integer
constraints. Satisfaction of state constraints under the projected input sequence is guar-
anteed by applying a robust reformulation to the original relaxed problem. Hence, the
proposed method is guaranteed to produce solutions satisfying both input as well as state
constraints.

We find the class of Pulse-Width Modulated (PWM) systems particularly well suited for
this method. We present a suitable projection function for these systems, and with exper-
imentations on different power converter topologies we show one can obtain substantially
enhanced control performances using our method with respect to a controller based on
plain linear MPC — at the same computational effort. The choice of the projection function is the key for good performance.

Finally, in order to reduce the conservatism of this approach, we extend it by adapting the idea of affine policies for the robust control of linear dynamic systems to the case of mixed-integer inputs. We illustrate the efficacy of this approach on a DC–DC buck power converter.

1.1 Contributions

Part I.

- The connection exposed in Theorem 3.2 is new. It is of practical importance because it does not depend on the way the dual is solved, and it enables us to design the solution scheme based on resource contraction. It is also important to know that the result holds in theory, because in some cases it is difficult to observe numerically, see the discussion in Section 5.2.1. Furthermore, the Assumption 3.1 required for the Theorem to hold (see counterexample A.4) sheds light on procedures to improve the numerics when solving problem instances affected by a high degree of symmetry.

It should be further emphasized that this theorem does not coincide with the (somewhat) known result stated in Theorem 6.1, which concerns structural properties of vertices of the convexified problem. In fact, they hold under totally different assumptions. Moreover, even though Theorem 6.1 seems to be known, as it is implied in [Bertsekas et al., 1983], we couldn’t find it explicitly stated anywhere in the literature.

- The solution scheme based on contractions proposed in Theorem 4.1 is new. It does not require any post processing of the inner solutions to achieve feasibility. The price to be paid is suboptimality, which however decreases in relative terms as the problems become larger. We certify this by providing performance bounds for the recovered solutions in Theorem 4.2, and in Theorem 4.4 which is valid under weaker assumptions. We also don’t need the assumption on the subproblem structure made in Theorem 3.1: our fundamental requirement becomes global feasibility of the contracted problem, implied in Assumption 3.1.

- It was known that it is possible to tighten the number of “problematic” subsystems from $m+1$, obtained using the Shapley–Folkman–Starr Theorem, to $m$, using an argument based on simplex tableaus. Beyond revisiting these results, our contribution here is to further tighten them, bounding the number of problematic subsystems by
the rank of the matrix of coupling constraints. We also show that this result can be further strengthened when additional structure on the coupling constraints matrix is present, such as in the case of optimization over certain network topologies, see Theorem 4.3.

- We propose a new control scheme based on optimization to achieve charging coordination of a fleet of Electric Vehicles (EVs). The controller organizes individual charging sequences such that both local requirements (e.g., a desired final state of charge) as well as global constraints (e.g., capacity limits of network links) are satisfied. It is designed to support both fixed as well as continuously-tunable charge rates, and is compatible with the way modern Li-ion batteries can be charged. It is also computationally viable: instances involving up to 10'000 EVs have been solved using our proposed methods within a few seconds ($\leq{}$ 10 sec) on a normal desktop PC. We also show how the scheme can be extended to handle different local models and requirements. For this, we study the case when V2G is available, i.e., when EVs batteries can inject power back in the network. It is furthermore possible to operate the proposed scheme in a centralized or distributed fashion: the former requires less communication iterations, while the latter allows for a higher degree of privacy.

- The use of the averaging method in [Anstreicher and Wolsey, 2009] to solve mixed integer problems is, to the best of our knowledge, new (Theorem 6.2). The method was developed for convex programs, but is not prominent because it is slow. Interestingly, this is not the case for the class of mixed integer problems studied in this part of the Thesis, because the integer variables “get stuck” during the dual iterations, and averaging exposes this quickly. The proposed method is also useful in practice: it allows us to solve difficult instances stemming from supply chain optimization quickly and reliably. For these, general purpose solvers are inadequate.

Part II.

- We studied the somewhat neglected case of endogenous uncertainties. We introduced a new uncertainty model that has combinatorial structure and derived its explicit robust counterpart. The proposed robustification procedure is computationally attractive. The use of this model for scheduling problems subject to uncertainty is new, and the application of these results to obtain flexible consumption schedules for industrial power consumers is also new.

- For the class of linear systems with mixed-integer input constraints, we propose a control scheme that combines an input rectification procedure with a constraint tightening approach. The proposed approach ensures feasibility in terms of both input as well as state constraints. We found the class of PWM systems particularly well suited for this technique: we provide a suitable projection function for this
case together with the recipe on how to calculate the necessary tightening. The performance of the proposed scheme is then tested on different power converter topologies. We observed substantial performance improvements over a plain MPC implementation – at the same computational effort.

- We extended the notion of affine recourse to the control of generic MLD systems subject to disturbances. The key idea proposed here is to optimize the integer part of the problem open loop, while the continuous decisions are subject to affine recourse. We provide all the necessary explicit robust counterparts for this, and test the scheme for the control of a DC-DC buck power converter. The proposed method substantially outperforms an MPC controller that employs standard affine policies on a linearized model of the converter.

1.2 Publications

The material presented in this Thesis is based on the following works done in collaboration with colleagues.

Part I.


Part II.

- R. Vujanic, S. Mariethoz, P. Goulart and M. Morari, Robust Integer Optimization and Scheduling Problems for Large Electricity Consumers, *American Control
1 Introduction

Conference, 2012 [Vujanic et al., 2012].


Part I

Convex Approximations based on Lagrangian Duality, for the Optimization of Large Scale Systems
2. Introduction to the Part and Outline

In this part of the Thesis we investigate mixed-integer optimization problems in the form

\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in I} c_i^T x_i \\
\text{subject to} & \quad \sum_{i \in I} H_i x_i \leq b \\
& \quad x_i \in X_i \quad \forall i \in I.
\end{align*}
\]

(\(P\))

We refer to \(b \in \mathbb{R}^m\) as the resource vector, and to the sets \(X_i\) as the subsystems. We assume that each of the sets \(X_i\) is a non-empty, compact, mixed-integer polyhedral set that can be written as

\[X_i = \{ x \in \mathbb{R}^q \times \mathbb{Z}^n_i \mid A_i x \leq d_i \},\]

with \(A_i \in \mathbb{R}^{m_i \times n_i}\) and \(d_i \in \mathbb{R}^{m_i}\). We further assume that the problem \(P\) is feasible and that the total number of subsystems \(|I|\) is greater than the length \(m\) of the resource vector. Our principal interest is in large-scale optimization problems, i.e. those for which \(|I| \gg m\), while remaining finite.

Problem \(P\) can be viewed generically as modeling any problem for which a large number of subproblems defined on the domains \(X_i\), whose description can include integer variables, are coupled through a small number of complicating constraints \(\sum_{i \in I} H_i x_i \leq b\). Simple examples of problems in this form include classical combinatorial programs such as the multidimensional knapsack problem, in which \(X_i = \{0,1\}\), and \(c_i \geq 0, H_i \geq 0\) [Wilbaut et al., 2008].

More complicated instances of problems in the form \(P\), with more detailed models for the subsystems \(X_i\), arise in a variety of contexts. In power systems, scheduling operations of power generation plants [Yamin, 2004] is a decision problem in which the subsystems are the generating units, integer variables in the local models arise due to, e.g., start-up and shut-down costs, and the coupling constraints are related to the requirement that generation must match load. In supply chain management, models fitting \(P\) appear in the problem of partial shipments [Dawande et al., 2006]. Portfolio optimization for small investors, for which mixed-integer models have been proposed, is another example application [Baumann and Trautmann, 2013]. Finally, some sparse problems that do not naturally possess the structure of \(P\) can be reformulated to fit our framework by
appropriately permuting rows and columns of the constraints matrix; [Bergner et al., 2011] proposes a method to automate this procedure.

A direct solution of $P$ is typically problematic when the problem is very large, since the problem amounts to a mixed-integer linear program of possibly very large size. As a result, the Lagrange dual of $P$ is often taken as a useful alternative, because the resulting dual problem is separable in the subsystems despite the presence of the complicating constraints. When this dual problem is solved by an iterative method, e.g. using the subgradient method [Bertsekas, 1999], a candidate (primal) solution to $P$ can be computed at each iteration.

One of the major drawbacks of this approach is that, for problems affected by a non-zero duality gap such as $P$, any guarantee about the properties of these candidate primal solutions is lost. Even at the dual optimal solution, the associated candidate primal solutions may be suboptimal and can even be infeasible. It is for this reason that Lagrangian duality is typically not used to recover primal solutions directly, but rather as a mean to obtain lower bounds to the optimal objective value.

The principal goal of our work in this part of the Thesis is to propose a new solution method for problem $P$ that preserves the attractive features of solution via the Lagrange dual, while at the same time protecting the recovered primal solutions from infeasibiltity.

**Known Results and Current Contribution.** Lagrangian relaxation for mixed integer programs was first introduced by [Held and Karp, 1970], and many of its theoretical properties were described in [Geoffrion, 1974]. Properties of the inner solutions in the convex case are well known [Rockafellar, 1997a, Thm. 28.1]. It is also well known that in general these properties are lost in the mixed-integer case [Bertsekas, 1999, Section 5.5.3]. Duality for problems specifically in the form $P$ has been studied at least as early as in [Aubin and Ekeland, 1976], where some of its special features were first characterized. In particular, it was noted that the duality gap for this program structure vanishes as the problem increases in size, as measured by the cardinality of $I$. We will show that the mechanism behind this vanishing gap effect can also be used to recover “good” primal solutions for the mixed-integer program $P$ directly from the dual, in a way that resembles the convex (zero gap) case.

In practical applications, vanishing duality gap has been observed in [Bertsekas et al., 1983] in the context of unit commitments for power systems. In this case it is exploited in an algorithm that provides solutions to the extended master problem, but no connection to the solutions of the inner problem is provided. It also appears in the multistage stochastic integer programming literature [Birge and Dempster, 1996, Caroe and Schultz, 1999], where it is used to gauge the strength of the Lagrangian dual, but in which no relations to primal solutions are drawn. Another domain in which vanishing gap has been used is in communications, more precisely in optimization of multicarrier communication
systems [Yu and Lui, 2006]. However, in this case non-convexity is in the objective function rather than due to the presence of integer variables.

In this part of the Thesis we further investigate duality for programs structured as $\mathcal{P}$ and focus on the primal solutions recovered at the dual optimum. On the theoretical side, the contributions of this work are as follows:

1. A new relation between the optimizers of a convexified form of $\mathcal{P}$ and the solutions to the inner problem is provided. This relation holds under mild conditions that are commonly satisfied in practice.

2. In light of this relation, we propose a method which guarantees that the solutions recovered from solving a modified dual are feasible for the original problem $\mathcal{P}$.

3. An upper bound on the optimality gap of the solutions recovered using this method is also provided.

4. The previous results assume that an exact optimal dual solution is available. In Chapter 6 we extend them by proposing a solution method for $\mathcal{P}$ with convergence guarantees, borrowing ideas and results from the convex literature. The scheme is based on averaging.

From an application perspective, we note that the proposed procedures are straightforward to implement and are amenable to distributed computation. Further, our performance bound indicates that the quality of the feasible solutions recovered improves as the problem size increases. Finally, the generality of the description of the subsystems $X_i$ allows our results to be exploited in a wide variety of applications. We show that the theoretical results are effective in practice via extensive numerical experiments on difficult problems stemming from the fields of power systems and supply chains. Our methods substantially outperform commercial solvers on these problems.

**Outline.** This part of the Thesis is structured as follows: in Chapter 3 we review some of the known results concerning duality for the specific structure of $\mathcal{P}$, and we provide a new result related to the primal solutions recovered from the dual. In Chapter 4 we propose a new method for primal solution recovery, and provide performance bounds for these solutions. We also give some results on how to further improve the solutions’ quality in some special cases, for instance for optimization problems over certain network topologies. In Chapter 5 we verify the efficacy of our proposed method on a difficult optimization problem stemming from power systems. We then study convergence behaviour in Chapter 6, i.e., how inner solutions behave when the dual iterate $\lambda^{[k]} \to \lambda^\ast$. We provide a solution method with convergence guarantees, and in the subsequent Chapter 7 we illustrate its effectiveness on difficult, industrial sized supply chain problems.

**Notation.** Given some optimization problem $\mathcal{A}$, we denote with $J_\mathcal{A}$ its optimal objective
and with $J_A(x)$ the performance of the solution $x$ with respect to the objective of $A$. For a given set $X$, we denote by $\text{conv}(X)$ its convex hull and by $\text{vert}(X)$ the set of vertices of $\text{conv}(X)$. The inequality “$\geq$” between vectors or matrices is always intended component-wise, and with $\otimes$ we indicate the cartesian product of sets. The support of a vector $\text{supp}(x)$ is the set of indexes of the non-zero elements: $\text{supp}(x) = \{i : x_i \neq 0\}$, while $(x)^+$ is the projection of $x$ onto the positive orthant, i.e., $(x)^+ = \max(0, x)$. We indicate with $|I|$ the cardinality of the set $I$. For the specific structure of $P$, we use the overbar symbol to indicate quantities related to the contracted version of $P$, as introduced in Section 4. Thus, for instance, $\overline{P}$ is the contracted form of $P$ and $\overline{D}$ is its dual. We use parenthesis to avoid confusing the sub- and superscripts, e.g., we denote by $(x_P)_i$ the part of $x_P$ related to subproblem $i \in I$ of problem $P$. In the algorithms, we use brackets to indicate iterations, e.g., $\lambda^{[k]}$ is the value of the variable $\lambda$ at iteration $k$. With $U[a, b]$ we denote the uniform distribution between $a$ and $b$. 
Consider the dual function $d : \mathbb{R}^m \to \mathbb{R}$ of problem $\mathcal{P}$, defined as

$$d(\lambda) \doteq \min_{x \in X} \left( \sum_{i \in I} c_i^T x_i + \lambda^T \left( \sum_{i \in I} H_i x_i - b \right) \right).$$

This function is known to provide lower bounds to the optimal objective of $\mathcal{P}$, i.e., $d(\lambda) \leq J_\mathcal{P}$ for all $\lambda \geq 0$. We are then interested in the best (largest) lower bound duality can provide, which is why we pose the following dual problem

$$\left\{ \begin{array}{l}
\sup_{\lambda} -\lambda^T b + \sum_{i \in I} \min_{x_i \in X_i} \left( c_i^T x_i + \lambda^T H_i x_i \right) \\
\text{s.t.} \quad \lambda \geq 0.
\end{array} \right. \quad (D)$$

We call $D$ the dual problem of $\mathcal{P}$, and we refer collectively to the minimizations within $D$, i.e.,

$$\min_{x_i \in X_i} \left( c_i^T x_i + \lambda^T H_i x_i \right), \quad (3.1)$$

as the inner problem. There is substantial practical interest in understanding the properties of the solutions to the inner problem (3.1) because they are obtained by solving $|I|$ independent (and lower dimensional) minimization problems, in contrast to the single large coupled problem $\mathcal{P}$. Additionally, they are usually obtained as by-products of methods used to solve $D$ (e.g. the subgradient method). These solutions, in particular those attained at the vertices of $\text{conv}(X_i)$, are the central object of this work:

**Definition 3.1** (inner problem solutions). For a given multiplier $\lambda \geq 0$, the set $X_i(\lambda) \subseteq \mathbb{R}^n$ is defined as the set of inner solutions that are attained at the vertices of $X_i$, i.e.

$$X_i(\lambda) \doteq \text{vert}(X_i) \cap \arg \min_{x_i \in X_i} \left( c_i^T x_i + \lambda^T H_i x_i \right). \quad (3.2)$$

Furthermore, we denote by $x(\lambda)$ any selection from the set $X(\lambda)$, and refer to it as an inner solution.

**Fact 3.1.** The sets $X_i(\lambda)$, $i \in I$, are non-empty for any $\lambda \geq 0$.

**Proof.** See Appendix C.1. \qed
3.1 Bound on Duality Gap

For a general mixed integer linear program, the inner solutions \( x(\lambda^*) \in \mathcal{X}(\lambda^*) \), in which \( \lambda^* \) is an optimizer of \( \mathcal{D} \), do not possess any “nice” property in general: they can be non-unique, suboptimal and even infeasible. We provide an example illustrating these issues in Appendix A.1. In this work we show that inner solutions for programs structured specifically as \( \mathcal{P} \) do acquire some useful properties. Informally speaking, these additional properties arise mainly from the fact that, as \( \mathcal{P} \) grows in size, it tends to closely approximate a convex program. One known result of this is that the duality gap between \( \mathcal{P} \) and \( \mathcal{D} \) vanishes, in relative terms, as \( |I| \) increases.

**Theorem 3.1** (bound on duality gap). Assume that for any \( x_i \in \text{conv}(X_i) \), there exists an \( \bar{x}_i \in X_i \) such that \( H_i \bar{x}_i \leq H_i x_i \). Then
\[
J_P - J_D \leq (m + 1) \max_{i \in I} \gamma_i, \quad \gamma_i = \max_{x_i \in X_i} c_i^T x_i - \min_{x_i \in X_i} c_i^T x_i. \tag{3.3}
\]

In consideration of Theorem 3.1, let \( |I| \) increase, while \( m \) remains constant and the the sets \( \{X_i\}_{i \in I} \) are uniformly bounded. If \( J_P \) increases linearly with \( |I| \), then
\[
\frac{J_P^*}{J_P^*} \rightarrow 0 \quad \text{as} \quad |I| \rightarrow \infty. \tag{3.4}
\]

An early proof of this result appears in [Aubin and Ekeland, 1976], while a more recent version is in [Bertsekas, 1996, Prop. 5.26, p. 374]. The same result also holds for more general problems; see [Bertsekas, 2009, Prop. 5.7.4, p. 223].

Note that while Theorem 3.1 ensures the existence of a primal feasible solution satisfying the performance bound (3.3), it does not provide an algorithmic way to produce it. Furthermore, the assumption required by Theorem 3.1 is restrictive; an example that does not fulfill this assumption is discussed in Chapter 5, see Remark 5.1. In this Thesis we lift this assumption, at the cost of conservatism and thus performance of the solutions recovered.

3.2 Geometric Properties of the Inner Solutions \( x(\lambda^*) \)

Here we present a new connection between the inner solutions \( x(\lambda^*) \) and the optimizers of the following optimization program
\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in I} c_i^T x_i \\
\text{subject to} & \quad \sum_{i \in I} H_i x_i \leq b \\
& \quad x_i \in \text{conv}(X_i) \quad \forall i \in I,
\end{align*}
\]
which amounts to a linear program. We denote by $J_{\mathcal{P}_{LP}}$ its optimal value, and by $x^*_{\mathcal{P}_{LP}}$ one of its optimizers. The relaxation $\mathcal{P}_{LP}$ plays a central role in Lagrangian duality for mixed integer programs; it is in fact well known that $\mathcal{P}_{LP}$ satisfies the (non-obvious) relation $J_{\mathcal{P}_{LP}} = J_D$ [Geoffrion, 1974, Thm. 1b, p.87]. Accordingly, $\mathcal{P}_{LP}$ is often used to gain insight into the strength of the relaxation, i.e., the tightness of the lower bounds to $J_{\mathcal{P}}$ provided by the Lagrangian dual. While in most practical cases one cannot solve $\mathcal{P}_{LP}$ directly since an explicit description of the polyhedral sets $\text{conv}(X_i)$ is required, column generation techniques construct approximations of $\mathcal{P}_{LP}$ [Barnhart et al., 1998, Desrosiers and Lübbecke, 2005, Vanderbeck, 2005]. It must be further emphasized that even though $\mathcal{P}_{LP}$ is a relaxation of $\mathcal{P}$ and is a linear program, it does not coincide with the standard linear relaxation in which the integrality constraints on the discrete variables are relaxed to intervals. In fact, $\mathcal{P}_{LP}$ is usually tighter; see [Geoffrion, 1974, Thm. 1a].

In consideration of the Shapley–Folkman–Starr theorem [Aubin and Ekeland, 1976, p.233], one can expect the vertices of the convexified problem $\mathcal{P}_{LP}$ to have “structure”, i.e. for $(x^*_{\mathcal{P}_{LP}})_i$ to belong to $X_i$ for at least $|I| - m - 1$ subproblems (and $(x^*_{\mathcal{P}_{LP}})_i \in \text{conv}(X_i)$ for the remaining $m + 1$ ones). This number can be improved to $|I| - m$ using an argument based on simplex tableaus instead of the Shapley–Folkman–Starr theorem. We use this tighter version here, and in the following new result, the crucial technical theorem of our work in this part, we extend it by establishing that the subproblems for which $(x^*_{\mathcal{P}_{LP}})_i \in X_i$ also “freeze” the corresponding inner solutions $x_i(\lambda^*)$.

**Assumption 3.1** (uniqueness for $\mathcal{P}_{LP}$ and $\mathcal{D}$). The programs $\mathcal{P}_{LP}$ and $\mathcal{D}$ have unique solutions $x^*_{\mathcal{P}_{LP}}$ and $\lambda^*$, respectively.

**Theorem 3.2.** Under Assumption 3.1, the solutions $x^*_{\mathcal{P}_{LP}}$ and $x(\lambda^*)$ differ in at most $m$ subproblem components, for any selection of $x(\lambda^*) \in \mathcal{X}(\lambda^*)$. That is, for all $x(\lambda^*) \in \mathcal{X}(\lambda^*)$ there exists $I_1 \subseteq I$, with $|I_1| \geq |I| - m$, such that $x_i(\lambda^*) = (x^*_{\mathcal{P}_{LP}})_i$.

**Proof.** See Appendix C.2. 

Assumption 3.1 concerns two linear programs (see program $\mathcal{P}_{LP}$ in Section C.2 for the LP version of $\mathcal{D}$). Uniqueness of primal and dual optimizers in the linear programming case is discussed in [Mangasarian, 1979], where necessary and sufficient conditions are provided. There are degenerate cases in which this assumption may fail, in particular when the problem’s data is affected by a high degree of symmetry. These cases, however, can always be avoided by adding negligible perturbations to the cost and resource vectors.

**Remark 3.1.** Theorem 3.2 holds even when the objective and the coupling constraints functions are concave. This is immediate by noticing that, in either case, local solutions are found at the vertices of $X_i$, according to a more general version of the Fundamental Theorem of Linear Programming, see [Bertsekas, 2009, Prop. 2.4.2]. The passage (C.1)
in the proof of Lemma 3.1 remains unchanged, and the proof of Theorem 3.2 follows verbatim.
Figure 3.1: Difference between the relaxation schemes discussed in the Thesis, illustrated on a purely binary problem. In (a) the constraints in the original problem setup are illustrated. This is the only problem considered in this Figure in which the set of points contained in the feasible set is finite. Figure (b) shows the traditional linear (LP) relaxation: it is the simplest to implement because one just drops the integrality requirements, but it also provides the worst approximation of the convex hull of the feasible set of $\mathcal{P}$, which is depicted in (d). The Lagrangian relaxation is a sort of middle ground. To see this, observe that the optimal value of the linear program $\mathcal{P}_{LP}$ depicted in (c) attains the same value of the lagrangian dual optimum. Notice how the $\mathcal{P}_{LP}$ relaxation in (c) is tighter than the relaxation in (b).
4. A Distributed Solution Method based on Resource Contraction

Informally speaking, Theorem 3.2 says that the inner solutions $x(\lambda^*)$ nearly coincide with those of $x_{LP}^*$, with the cardinality of their difference bounded by $m$, i.e., the dimension of the coupling constraint. Since $x_{LP}^*$ is feasible with respect to the coupling constraints and attains a better objective than $J_P^*$, one can expect the solutions obtained from solving the dual to be nearly feasible and to attain good objective values. In this section we exploit this result to propose a method aimed at obtaining “good” feasible solutions to problem $P$ in a distributed fashion.

4.1 Resource Contraction Method

Our proposed method is to contract the resources vector $b$ by an appropriate amount, which is determined by the results of the previous section. We show that any inner solution recovered at the dual optimizer $\lambda^*$ of the contracted problem is a feasible solution for $P$. We also provide a performance bound for these solutions, which indicates that their quality improves with increasing problem size.

Consider the following modified version of problem $P$

\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in I} c_i^T x_i \\
\text{subject to} & \quad \sum_{i \in I} H_i x_i \leq \bar{b} \\
& \quad x_i \in X_i \quad \forall i \in I.
\end{align*}
\]

$P$

The resource vector $b$ has been contracted to $\bar{b} = b - \rho$, where the $k$-th element of the contraction $\rho \in \mathbb{R}^m$ is given by

\[
\rho^k = m \cdot \max_{x_i \in X_i} \left( \max_{x_i \in X_i} H_i^k x_i - \min_{x_i \in X_i} H_i^k x_i \right), \quad (4.1)
\]

where $H_i^k$ is the $k$-th row of $H_i$. Correspondingly, we introduce the problems $P_{LP}$ and $\bar{B}$, defined similarly to $P_{LP}$ and $\bar{P}$, replacing the resource vector $b$ with $\bar{b}$. We next establish that the primal solutions recovered from the dual of $P$ are feasible for $P$. 

**Theorem 4.1** (feasible solutions). If Assumption 3.1 holds for the programs $\overline{P}_{\text{LP}}$ and $\overline{D}$, then any selection $x(\bar{\lambda}^*) \in \mathcal{X}(\bar{\lambda}^*)$ is feasible for $\overline{P}$, where $\bar{\lambda}^*$ is the optimal solution of $\overline{D}$.

**Proof.** See Appendix C.3.

The method is easy to implement because the amount of contraction required usually necessitates only simple computations, and these can be carried out in a distributed fashion. Furthermore, for the solution of the dual problem well-established methods exist (e.g., the subgradient method) and they can be directly applied here. In the Appendix Section A.3 we provide an analytical example that illustrates the theorem, while in Section A.4 we show how the theorem may fail in absence of Assumption 3.1.

In the next Theorem we assess the performance of the solutions $x(\lambda^*)$. In order to obtain an explicit bound for the optimality gap, we first make the following assumption on the existence of a Slater point with increasing slack.

**Assumption 4.1** (Slater point with increasing slack). There exist $\zeta > 0$ and $\bar{x}_i \in \text{conv}(X_i)$ for all $i \in I$ such that

$$\sum_{i \in I} H_i \bar{x}_i \leq \bar{b} - \zeta |I|.$$  \hspace{1cm} (4.2)

**Theorem 4.2** (performance guarantee). Suppose that the programs $\overline{P}_{\text{LP}}$ and $\overline{D}$ satisfy Assumption 3.1 and Assumption 4.1 holds. Then any solution $x(\bar{\lambda}^*) \in \mathcal{X}(\bar{\lambda}^*)$ recovered satisfies

$$J_P(x(\bar{\lambda}^*) ) - J_P^* \leq (m + \|\rho\|_{\infty} / \zeta) \cdot \max_{i \in I} \gamma_i,$$  \hspace{1cm} (4.3)

where $\gamma_i$ and $\rho$ are as defined in (3.3) and (4.1), respectively.

**Proof.** See Appendix C.4.

In view of Theorem 4.2, if the sets $\{X_i\}_{i \in I}$ are uniformly bounded and $J_P^*$ grows linearly in terms of $|I|$, then

$$\frac{J(x(\bar{\lambda}^*)) - J_P^*}{J_P^*} \to 0 \quad \text{as} \quad |I| \to \infty.$$  \hspace{1cm} (4.4)

Accordingly, the quality of the solutions recovered increases the larger the problem becomes, as the optimality gap decreases at a “$1/|I|$” rate. This behavior is similar to the behavior of the duality gap (3.4). In Section 4.3 we will discuss Assumption 4.1 and show that this behavior can be expected even in the absence of a Slater point.
Notice that reducing the available resources has the general effect of increasing the associated multipliers, i.e., $\|\hat{\lambda}^*\|_1 \geq \|\lambda^*\|_1$. In a context in which dual optimizers are interpreted as prices, the contraction can be seen as the necessary price uplift required to ensure that the “market participants” converge to a solution that is at least globally feasible, despite the presence of local non-convexities.

### 4.2 Reducing Conservatism

The contraction proposed in Theorem 4.1 can be interpreted as a robustification of problem $P$ toward alterations of $m$ local solutions $x_i$. In this section we take a closer look at the coupling constraints matrix $H = [H_1, H_2, \ldots, H_m]$ and discuss some special cases in which its structure can be exploited to safely reduce the necessary contraction.

Suppose that the matrix $H$ has block structure, as depicted in Figure 4.1(a). We introduce the set $I_k$ as the index set of the subsystems contributing to the $k$-th coupling constraint, i.e., for which $H^k_{ij} \neq 0$. As illustrated, we also define the submatrix $[H_i]_{i \in I_k}$, obtained by collecting the columns of $H$ related to the subsystems in $I_k$.

Such a block structured $H$ may arise in applications of optimization over tree or tree-star networks, as shown on Figure 4.1(b). In this case, the uniform contraction proposed in Theorem 4.1 can be safely reduced.

**Theorem 4.3.** Theorem 4.1 holds with the contraction (4.1) substituted by

$$\rho^k = \text{rank}([H_i]_{i \in I_k}) \cdot \max_{i \in I_k} \left( \max_{x_i \in X_i} H^k_{ij} x_i - \min_{x_i \in X_i} H^k_{ij} x_i \right). \quad (4.5)$$

*Proof. See Appendix C.5.*

This theorem implies, as a special case, that we can generally substitute $m$ with $\text{rank}(H)$ in (4.1), independently of whether the problem has block structure. This is important when the vectors determining the coupling constraints are linearly dependent. Examples exploiting this result are discussed in Chapters 5 and 7.

Furthermore, instead of immunizing against $\text{rank}([H_i]_{i \in I_k})$ times the largest subproblem budget consumption change, it is sufficient to immunize against the $\text{rank}([H_i]_{i \in I_k})$ largest ones, i.e.,

\[\rho^k = \alpha \cdot 1\] for some $\alpha \in \mathbb{R}_+$, as is often the case in practice, then $\|\hat{\lambda}^*\|_1 \geq \|\lambda^*\|_1$ follows immediately.

---

\[\text{To see this, note that } \hat{d}(\lambda) = d(\lambda) + \rho^T \hat{\lambda} \text{. Since } \hat{\lambda}^* \in \arg \max_{\lambda \geq 0} \hat{d}(\lambda), \text{ we have that } d(\lambda^*) + \rho^T \lambda^* \geq d(\lambda) + \rho^T \hat{\lambda} \text{ for all } \lambda \geq 0, \text{ and hence in particular also for } \lambda^*. \text{ On the other hand } d(\lambda^*) \geq d(\lambda) \text{ for all } \lambda \geq 0. \text{ Using these relations, we obtain } \hat{d}(\lambda^*) + \rho^T \hat{\lambda} \geq (\lambda^*) + \rho^T \lambda^* \geq (\hat{\lambda}^*) + \rho^T \hat{\lambda} \text{, and thus that } \rho^T \hat{\lambda} \geq \rho^T \lambda^*. \text{ If } \rho = \alpha \cdot 1 \text{ for some } \alpha \in \mathbb{R}_+, \text{ as is often the case in practice, then } \|\hat{\lambda}^*\|_1 \geq \|\lambda^*\|_1 \text{ follows immediately.} \]
Subsystems

\[ H = \begin{bmatrix}
  1 & 6 & 7 & 13 & 14 & 19 & 20 & 23 & 24 & 28 \\
 7 & 13 & 14 & 19 & 23 & 20 & 24 & 28 & - & - \\
\end{bmatrix}
\]

\[ [H_i]_{i \in I_k} \]

(a)

(b)

\[ \rho^k = \max_{i \in I_k} \left( \sum_{i \in I} \max_{x_i \in X_i} \left( \min_{x_i \in X_i} H^k_i x_i \right) - \min_{x_i \in X_i} H^k_i x_i \right). \]  \hspace{1cm} (4.6)

Finally, an important subclass of problems for which we can suppress the necessary contraction to \( \rho = 0 \) is the following.

Remark 4.2. If \( H_i x_i \geq 0 \) for all \( x_i \in X_i \), and \( 0 \in X_i \), then one can obtain the same performance bound as in (4.3) while setting \( \rho = 0 \), and a feasible solution can be recovered by setting \( x_i(\lambda^*) = 0 \) for at most \( m \) subsystem solutions.

This is for instance the case for the (multidimensional) knapsack problem and some of its variants. Namely, a feasible solution is obtained by removing at most \( m \) items from the knapsacks. The supply chain application discussed in Chapter 7 also satisfies this assumption. Since conservatism is eliminated in this case, the primal solutions recovered are nearly optimal.
4.3 Further Discussion on the Performance Bound

One of the key factors contributing to the optimality gap identified in Theorem 4.2 is the performance loss due to the contraction $\rho$, determined by $[J_{P_{LP}}(0) - J_{P_{LP}}(\epsilon)]$; see the proof of Theorem 4.2, in particular the term (ii), in Section C.4. In Theorem 4.2, Assumption 4.1 allows us to establish an explicit bound on this term. Here we show that this performance loss can be characterized by the data of only $m$ subsystems, which explains why one may expect a behavior for the optimality gap similar to (4.4) even in the absence of Assumption 4.1.

**Proposition 4.1.** Consider the perturbed version of the program $P_{LP}$

\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in I} c_i^T x_i \\
\text{subject to} & \quad \sum_{i \in I} H_i x_i \leq b + \varepsilon 1 \\
& \quad x_i \in \text{conv}(X_i) \quad i \in I,
\end{align*}
\]

whose optimal value is denoted by $J_{P_{LP}}(\varepsilon)$. Let $D_i = (X_i, H_i, c_i)$ be the tuple representing the data of the $i^{th}$ subsystem, where the sets $X_i$ are all compact. Then, there exist a partition $I = I_1 \cup I_2$ and a constant $L(I_2) = L\left(\{D_i\}_{i \in I_2}\right)$, only depending on the data of subsystems indexed by $I_2$, such that $|I_2| \leq m$ and

\[0 \leq J_{P_{LP}}(0) - J_{P_{LP}}(\varepsilon) \leq L(I_2)\varepsilon, \quad \forall \varepsilon \in \mathbb{R}_+.
\]

**Proof.** The proof, along with some preliminaries, is in Appendix C.6. \qed

This result allows us to provide the following performance bound on the optimality gap for the recovered solutions.

**Theorem 4.4.** Suppose the programs $P_{LP}$ and $D$ satisfy Assumption 3.1. Then, any solution $x(\bar{x}^*) \in \mathcal{X}(\bar{x}^*)$ recovered satisfies

\[J_P(x(\bar{x}^*)) - J_P \leq m \cdot \max_{i \in I} \gamma_i + \max_{\|\rho\|_{\infty} \leq m} L(I_2) \cdot \|\rho\|_{\infty}, \quad (4.7)
\]

where $\gamma_i$ and $\rho$ are as defined in (3.3) and (4.1), respectively, and $L(I_2)$ is the constant determined by subsystems indexed by $I_2$ as introduced in Proposition 4.1.

The proof of Theorem 4.4 essentially follows the same analysis of Section C.4. In light of this theorem, it is then clear that if $\{\gamma_i\}_{i \in I}$ and $\{L(I_2)\}_{b \in I}$ are uniformly bounded, and $J_P$ grows linearly with $|I|$, we reach the same conclusion on the optimality gap behavior as in (4.4). These uniform bounds are satisfied if the diversity of the subsystems added to the problem, when we increase its size, is limited.
A major new source of demand for future electricity distribution systems is the increasing use of hybrid or fully electric vehicles (EVs). Recent studies project EVs sales to constitute between 18% and 45% of total car sales in the US by 2020 [Becker et al., 2009]. This transition will cause a substantial change in the aggregated electric load profiles, with serious repercussions on the distribution networks, as highlighted by several studies [Clement-Nyns et al., 2010, Taylor et al., 2010].

In order to ensure the ongoing reliable operation of electricity distribution networks, control mechanisms must be put in place to manage charging of electric vehicles. To achieve this using a centralized control architecture, a distribution system operator (DSO) would have to handle a potentially massive number of EVs. To avoid this difficulty, the control structure that is expected to prevail is a hierarchical one [Callaway and Hiskens, 2011, Sioshansi, 2012]. In this setting so-called aggregators will provide the necessary interface, acting as virtual power plants from the perspective of the DSO or, generically, towards the higher levels of the control hierarchy, while managing the load fleet under their jurisdiction. In such a control architecture, the DSO will provide the limits on the available network resources that the aggregator can use to charge its fleet. These limits must be determined so as to ensure that network equipment is not damaged. In turn, the aggregator must decide on a charging schedule for its fleet that is compatible with these constraints.

In this Chapter we are concerned with the computations to be carried out at the aggregator level. One option to fulfil the above needs is to use computational methods based on mathematical optimization. Optimization can incorporate the diverse charging requirements of each individual EV, while ensuring that, in aggregation, the network limits are satisfied. In such a framework it is furthermore possible to define an objective function that will determine the best charging schedule among all possible schedules that satisfy the aforementioned local (EV level) and global (network level) constraints. We describe the type of optimization programs an aggregator has to solve. Owing to fixed charge rate requirements, integer variable will be used in the model, leading to a mixed-integer optimization problem formulation; such models have already appeared in
the literature [Di Giorgio et al., 2012, Ruiz et al., 2009]. We will show that, despite the non-convex nature of this formulation, we can get approximate solutions at very low computational cost.

This is achieved by deploying the contraction method of Chapter 4. It allows us to recover solutions that are guaranteed to be feasible with respect to the local and global constraints without the recourse to any external solver. We also verify that the quality of the solutions improves as the problem instances become larger.

In our simulation results we show that the computation times on a normal PC, even for these larger instances entailing 10'000 EVs, are very fast (≤ 10 sec). Furthermore, depending on the availability of computational power on the EV, it is optionally possible to carry the bulk of the required computations distributedly.

5.1 Problem Formulation and Mathematical Modeling

In this section we first describe the control problem from an aggregator viewpoint. We then propose to tackle this control task using optimization, and provide a mathematical model to this end.

5.1.1 Scenario and Control Task Description

We take the perspective of a load aggregator responsible for the electric vehicles (EVs) connected to a particular section of a radial distribution network, as depicted in Figure 5.1. Most of the EV charging occurs overnight, but left uncontrolled it typically happens as soon as the EVs are connected to the network. Since EVs are relatively large loads this may cause load spikes which, in the worst case, can cause a partial blackout of the distribution system [Clement-Nyns et al., 2010].

Our control task is to determine a charging schedule for each individual EV, such that the aggregated EV load is more evenly distributed over the entire night period. Additionally, the flows through a particular branch of the network under control must be kept within some limits. This may be for instance necessary to prevent an excessive voltage drop within the branch.

We assume that EV charging stations draw power at fixed rates, which is usually the case in practice [Di Giorgio et al., 2012]. Charging can be interrupted and resumed, but in order to avoid excessive switching, once charging starts it must continue for at least 20 minutes. This is a reasonable way of charging Lithium-Ion batteries, which are the most common in EVs, because they do not present memory effects and thus
charging interruption does not cause any appreciable degradation [Riezenman, 1995]. Non-interruptible charging is not discussed in this paper as it is uncommon in practice. However, those applications for which this is necessary (e.g., Nickel-Cadmium batteries) can be readily incorporated in our proposed framework with an appropriate design of the local constraints.

We therefore split the overnight period into intervals of 20 minutes each, and assume that the aggregator has the authority to flag, for each individual EV, the available charging time slots.

The information to be communicated between each EV and the aggregator depends on the mode in which our proposed method is implemented. We elaborate on this in Section 5.2. For the moment, let us assume that when the EV is connected, it transmits all its charging requirements to the aggregator, i.e., the initial state of charge (SOC), a desired minimum final SOC and the time when the EV is planned to be disconnected. The initial SOC is directly measured by the EV, while the other two parameters can be set to some default values, so that the EV user does not need to manually enter this information every time.

For the sake of simplicity, we also assume that all the EVs are connected at the time when the charging schedule is established. This assumption can be easily relaxed by buffering...
newly connected EVs, and recomputing every 20 minutes a charging schedule with the new population information.

5.1.2 Optimization Problem Model

We begin by encoding the control task described in the previous subsection as an optimization problem. Generally, such a model will comprise a large number of subsystems, with a global objective and in the presence of some coupling constraints. Owing to the hierarchical structure discussed previously, the aggregator is responsible only for a limited number of network constraints. Thus, the number of coupling constraints is typically significantly smaller than the number of the subsystems under control.

The optimization problem model we propose in this paper is as follows:

- **Decision Variables.** The binary variables $u_i[k] \in \{0, 1\}$ are the decision variables representing the flag that the aggregator assigns to time-slot $k$ for the electric vehicle $i$. When $u_i[k] = 1$, the EV is allowed to charge during the $k$-th time slot and, vice versa, when $u_i[k] = 0$ charging cannot occur.

- **Objective Function.** We model the requirement of evenly distributing load throughout the night as a reference tracking objective, in which the signal $P_{\text{ref}}$ is chosen so as to achieve the desired “valley fill”:

$$\min_{u_i} \sum_{k=0}^{N-1} \left| \sum_{i \in I} P_i u_i[k] - P_{\text{ref}}[k] \right|,$$

where $P_i$ is the power consumption of the $i$-th EV when charging. Alternatively, one can write (5.1) as

$$\min_{u_i, r} \sum_{k=0}^{N-1} r[k],$$

(5.2)

together with the additional constraint

$$-r[k] \leq \sum_{i \in I^c} P_i u_i[k] - P_{\text{ref}}[k] \leq r[k] \quad k \in \mathbb{N}_{[0,N-1]}$$

(5.3)

- **Coupling Constraints.** In addition to (5.3), coupling constraints arise from the limits on the flows through the critical branch. We model these as hard constraints on the consumption of the electric vehicles $i \in I^c \subseteq I$ belonging to that critical branch,

$$\sum_{i \in I^c} P_i u_i[k] \leq P_{\text{max}}[k] \quad k \in \mathbb{N}_{[0,N-1]}.$$
• **Subsystems Model.** The subsystems in this application are the EVs batteries that are to be charged. We denote by $e_i[k]$ the charge level of the $i$-th battery. The initial state of charge is $E_i^{\text{init}}$, and we require that the state of charge reaches $E_i^{\text{ref}}$ by $N_i \leq N$, the time when the vehicle is planned to be unplugged. The charging conversion efficiency is $\zeta_i < 1$, and the battery’s maximum capacity is $E_i^{\text{max}}$. Loss of charge when not in use is neglected, as it is typically a small quantity [Riezenman, 1995]. Subsystems are then modeled as follows:

$$
e_i[0] = E_i^{\text{init}}$$

$$
e_i[k + 1] = e_i[k] + (P_i \Delta T \zeta_i) u_i[k] \quad k \in \mathbb{N}_{[0, N-1]}$$

$$
e_i[N_i] \geq E_i^{\text{ref}}$$

$$
e_i[k] \leq E_i^{\text{max}} \quad k \in \mathbb{N}_{[0, N]}$$

$$
u_i \in \{0, 1\}^N.$$

The optimization problem that is to be solved at the aggregator level amounts then to

$$\begin{aligned}
\text{minimize} & \quad \text{reference tracking error (5.2)} \\
\text{subject to} & \quad \text{tracking error definition (5.3)} \\
& \quad \text{critical branch flow constraint (5.4)} \\
& \quad (e_i, u_i) \in X_i,
\end{aligned}$$

(5.6)

in which

$$X_i = \left\{ \begin{bmatrix} e_i \\ u_i \end{bmatrix} \in (\mathbb{R} \times \mathbb{Z})^N \mid \begin{array}{c}
e_i[k] = E_i^{\text{init}} + \sum_{t=0}^{k} B_i u_i[t] \\
e_i \leq E_i^{\text{max}} \\
e_i[N_i] \geq E_i^{\text{ref}} \\
0 \leq u_i \leq 1\end{array} \right\},$$

(5.7)

where $B_i \triangleq P_i \Delta T \zeta_i$.

Note that problem (5.6) fits the structure of $\mathcal{P}$, where the coupling constraints are for instance the branch flow constraints (5.4), and the resource is the power line capacity. The subsystems are the EV batteries, and the local model $X_i$ is the charging model (5.7). This may integer variables. In the EV aggregator application, these arise due to fixed rate charging. In other smart grid applications, integer variables can be used to model on/off devices such as thermostatically controlled loads [Callaway and Hiskens, 2011].

**Remark 5.1.** The assumption in Theorem 3.1 does not apply to this model. According to (5.4), $H_i x_i = P_i u_i$, and a fractional $x_i \in \text{conv}(X_i)$ implies that in at least one time step, charge is happening at a partial rate. To rectify it, one has to either increase it to the fixed charge rate or decrease it to 0. In the latter case it may however be necessary
to increase charging at another time step, in order to satisfy the energy requirement of the EV in (5.7). Since any such rectification will cause an increase of resources used at some time, the assumption cannot be met.

5.2 Application of the Method based on Resource Contraction

We now describe how to apply the method presented in Chapter 4 to the aggregator’s optimization problem (5.6). We also highlight several important practicalities.

5.2.1 Implementation and Practical Remarks

To make the description easier to follow, we have divided it in a number of separate steps.

Step 1 (resource contraction). The number of coupling constraints in (5.6) is $3N$. However, for any solution $x(\lambda)$ the feasibility of the coupling constraints (5.3) is guaranteed by increasing the value $r$, ensuring that these constraint need not to be contracted. Constraints on the flows through the critical branch are true hard constraints whose feasibility is the main objective of the tightening procedure (4.1) proposed in Chapter 4.

For these constraints, we note that the rank of the corresponding submatrix $[H_{i,k}]_{i,k}$ is $N$. Notice further that this number does not need to be increased if other non-overlapping critical branches are added to the problem model. Thus, for these networked problems, Theorem 4.3 allows for a substantial reduction in necessary conservatism.

We then have

$$\sum_{i \in I^d} P_i u_i[k] \leq \bar{P}^\max[k] = P^\max[k] - \rho, \quad \text{with} \quad \rho = N \cdot \max_{i \in I} P_i.$$  

(5.8)

Remark 5.2. The interpretation of this result is that in the non-contracted solution $x(\lambda^*)$ at most $N$ EVs can be responsible for hard constraint violations which, in the worst case, occur all at the same time step.

Step 2 (perturbation). In our simulations we have observed that adding small perturbations to the cost vector, as in

$$\min_{u_i} \sum_{k=0}^{N-1} \left( r[k] + \sum_{i \in I} \delta_i[k](P_i u_i[k]) \right)$$  

(5.9)
with small $\delta_i$, significantly enhances dual convergence; we recommend to always implement this.

Intuitively speaking, the reason why this helps is that the master problem intends to achieve coordination by issuing the “prices” $\lambda$, but each individual EV “sees” the same price without perturbation term. When the diversity of the subsystems $X_i$ is limited, i.e., the EVs have similar dynamics, their reaction to price profiles is also similar. Consequently, as the price profile is adjusted, they all respond similarly, leading to an oscillatory behaviour which is solved only if the dual solution $\lambda^*$ is found with very high precision. This effect has been already observed in the literature [Ma et al., 2013]. The perturbation term in (5.9) artificially changes the price received by each individual EV, neutralizing oscillations earlier and hence leading to substantially faster convergence rates when solving the dual. The addition of these perturbations also contributes towards the satisfaction of Assumption 3.1 for highly symmetric cases, in which the subsystems are very similar to each other.

**Step 3 (setup and solve dual problem).** Dualizing the coupling constraints in (5.6) leads to the dual function

\[
d(\lambda) = \sum_{i \in \mathcal{I}} \min_{x_i \in X_i} P_i u_i^T (\delta_i + \lambda_1 - \lambda_2 + \lambda_3 \cdot \mathbb{I}^c(i)) + \min_{r \geq 0} r^T (1 - \lambda_1 - \lambda_2) - (\lambda_1 - \lambda_2)^T P_{\text{ref}} - \lambda_3 \mathcal{B}_{\text{max}},
\]

in which $\lambda_1$ and $\lambda_2$ are the dual multipliers associated with (5.3), while $\lambda_3$ is related to the dualization of (5.4), and

\[
\mathbb{I}^c(i) = \begin{cases} 1 & \text{if } i \in \mathcal{I}^c \\ 0 & \text{otherwise}. \end{cases}
\]

Due to the term containing the minimization over $r \geq 0$, the dual function remains bounded only when $\lambda_1 + \lambda_2 \leq 1$. For these values, $r(\lambda) = 0$, and the dual function reduces to

\[
d(\lambda) = \sum_{i \in \mathcal{I}} \min_{x_i \in X_i} P_i u_i^T (\delta_i + \lambda_1 - \lambda_2 + \lambda_3 \cdot \mathbb{I}^c(i)) - (\lambda_1 - \lambda_2)^T P_{\text{ref}} - \lambda_3 \mathcal{B}_{\text{max}},
\]

which we want to maximize over

\[
\lambda \in \Lambda = \{ (\lambda_1, \lambda_3) \in \mathbb{R}^{2N} \mid \lambda \geq 0, \lambda_1 + \lambda_2 \leq 1 \}.
\]

Given an initialization point $\lambda^{[1]}$, the corresponding dual problem, as defined in $\mathcal{D}$, can be solved by deploying the projected subgradient method [Anstreicher and Wolsey, 2009], i.e.,

\[
\lambda^{[n+1]} = P_{\Lambda} \left( \lambda^{[n]} + \varepsilon^{[n]} \cdot \gamma^{[n]} \right),
\]
in which $\gamma[n] = \sum_{i \in I} H_i x(\lambda[n]) - b$ is a valid subgradient, and $s[n]$ is the chosen stepsize rule. Several rules exist which will guarantee the convergence $\lambda[n] \to \lambda^* \in \Lambda^*$, see [Anstreicher and Wolsey, 2009, Thm. 2]. Let us remark that the iterative update rule (5.11) involves a projection operation onto $\Lambda$ at each step. For $\Lambda$ as in (5.10), this projection is simple to compute.

**Step 4 (inner problem).** At each iteration we need to compute a solution to the inner problem $x(\lambda[n])$ in order to determine a valid subgradient $\gamma[n]$. The inner problem consists of $|I|$ decoupled local optimization programs of the form

$$\min_{(u, e) \in \mathcal{X}_i} P_i u_i^T (\delta_i + 2\lambda_1 - 1 + \lambda_3 \cdot \mathcal{G}(i)).$$

(5.12)

The method proposed in this paper is most useful when these minimizations are simple. This is the case for the EV aggregator application. In fact, for any given dual multiplier $\lambda \in \Lambda$, the solution to (5.12) — the optimal local charging policy $u_i^*(\lambda)$ — is **greedy**. As we illustrate in Appendix B, due to this greedy subproblem structure, the required local computation essentially amounts to a single vector sort, which is computationally extremely lightweight.

Note that these computations do not need to be executed by the aggregator itself. If computational power is available on the EVs to calculate solutions to (5.12), then the computational burden can be distributed. The advantage is that when the optimization problem is solved in this fashion, a higher degree of privacy results, as the EVs do not need to provide any local data to the aggregator (i.e., any of the parameters describing $X_i$), but only their planned consumption for a given price profile. The drawback is that such an iterative process will require more communication rounds to converge to a solution, instead of transmitting information only once.

### 5.2.2 Simulation Setup

We compare the performance of our proposed method with the results provided by CPLEX 12.5. For each fleet size considered, we generate 10 random instances based on the parameters provided in Table 5.2. In order to ensure a fair comparison, since CPLEX is generally unable to find exact solutions to the model (5.6), we first run our proposed algorithm on each problem instance, record the optimality gap (we get a tight lower bound for free as a by-product of our method), and then run CPLEX up to the same optimality gap. Furthermore, the perturbation $\delta_i$ is added to the objective function, and the perturbed problem is the one on which we deploy both our method as well CPLEX. This ensures that both methods are exposed to exactly the same problem. All our tests are performed on a Desktop PC with 8GB of RAM and a 3.10 GHz processor.
5 Application – Electric Vehicles

Aggregator

<table>
<thead>
<tr>
<th># PEVs</th>
<th>Tracking Error (%)</th>
<th>Solve time↑ (sec)</th>
<th>CPLEX Solve time↑↑ (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Avg</td>
<td>Max</td>
</tr>
<tr>
<td>50</td>
<td>9.94</td>
<td>16.85</td>
<td>23.17</td>
</tr>
<tr>
<td>100</td>
<td>4.23</td>
<td>5.25</td>
<td>6.09</td>
</tr>
<tr>
<td>200</td>
<td>1.45</td>
<td>1.87</td>
<td>2.48</td>
</tr>
<tr>
<td>500</td>
<td>0.85</td>
<td>0.97</td>
<td>1.08</td>
</tr>
<tr>
<td>1000</td>
<td>0.56</td>
<td>0.70</td>
<td>0.99</td>
</tr>
<tr>
<td>2000</td>
<td>0.29</td>
<td>0.40</td>
<td>0.56</td>
</tr>
<tr>
<td>5000</td>
<td>0.17</td>
<td>0.23</td>
<td>0.32</td>
</tr>
<tr>
<td>10000</td>
<td>0.13</td>
<td>0.16</td>
<td>0.21</td>
</tr>
</tbody>
</table>

(*) ≤ 0.3 sec (imprecise measurements).

↑sum of the time spent solving the dual iterations.


Table 5.1: Computational results for charging coordination problem.

5.2.3 Simulation Results

The numeric performance determined from these experiments is reported in Table 5.1. Solve times are stable and well within 10 seconds even for the largest instances. If needed, these can be further reduced by exploiting parallelism, since the bulk of the computations consists in determining inner solutions, or by rewriting the code using a faster language (e.g., C). Tracking performance is measured by

$$\theta = \frac{\|r(\lambda^{[\text{end}])}\|_1}{\|P_{\text{ref}}\|} \cdot 100,$$

where $r(\lambda^{[\text{end}])}$ is the tracking error of the solution recovered from the last dual iterate $\lambda^{[\text{end}]}$. This performance metric represents the achieved energy of the tracking error, relative to the total amount of reference energy to be drawn by the population.

Figure 5.2 represents the typical global performance plots for the primal solutions recovered through our contraction scheme, for an instance with $|I| = 5,000$ EVs. Figure 5.2(a) depicts the reference tracking performance at some intermediate and the final iteration. The reference signal is tracked well, and Figure 5.2(b) shows the resulting “valley fill” achieved. In Figure 5.4 we can observe the flow limits through the critical branch (dotted line), and their tightened counterparts (dashed). At the final iteration, the solution recovered satisfies the line capacity constraints. Figure 5.3 depicts the local charging pattern established by the system. It satisfies all the local constraints.
5 Application – Electric Vehicles

### Aggregator

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$</td>
<td>I</td>
</tr>
<tr>
<td>$</td>
<td>I^c</td>
<td>$</td>
</tr>
<tr>
<td>$P_i$</td>
<td>$[3 - 5]$</td>
<td>kW</td>
</tr>
<tr>
<td>$E_{\text{max}}$</td>
<td>$[8 - 16]$</td>
<td>kWh</td>
</tr>
<tr>
<td>$E_{\text{init}}^i$</td>
<td>$[0.2 - 0.5] \cdot E_{\text{max}}^i$</td>
<td>kWh</td>
</tr>
<tr>
<td>$E_{\text{ref}}^i$</td>
<td>$[0.55 - 0.8] \cdot E_{\text{max}}^i$</td>
<td>kWh</td>
</tr>
<tr>
<td>$\zeta_i$</td>
<td>$[0.015 - 0.075]$</td>
<td>–</td>
</tr>
<tr>
<td>$\Delta T$</td>
<td>20</td>
<td>min</td>
</tr>
<tr>
<td>$N$</td>
<td>24</td>
<td>–</td>
</tr>
<tr>
<td>$\delta_i$</td>
<td>$[0 - 0.03]$</td>
<td>–</td>
</tr>
<tr>
<td>$P_{\text{max}}$</td>
<td>$4 \cdot \sum_{i=0}^N E_{\text{ref}}^i - E_{\text{init}}^i$</td>
<td>kW</td>
</tr>
</tbody>
</table>

Table 5.2: Parameters used in the simulations. Values in the brackets are sampled from a uniform distribution.

In Figure 5.5 we report the typical primal and dual convergence behaviour. Figure 5.5(a) depicts the constraint violation of the primal solutions recovered, with and without the perturbation terms discussed in Section 5.2.1. Convergence with the small additive perturbations (cf. Table 5.2) is substantially enhanced.

### 5.3 Vehicle–to–Grid (V2G) Extension

In this section we illustrate the capability of the method to support different local models. For this, we consider the charging coordination problem when the batteries connected are also able to inject power back in the network, which is commonly referred to as vehicle-to-grid or V2G.

Such a functionality unfolds opportunities. During times of high peak demand, an electricity producer may for instance use a fleet of EV providing V2G to avoid the (expensive) activation of an additional generation unit. Another use case is in the context of microgrids. Here, V2G can be used to maintain a functional local grid independently of the main transmission network. This is for instance of interest for military entities, since a fleet of vehicles can be used to maintain a base operative in case of attacks aimed at disrupting the electricity supply.

On the other hand, there are also obstacles and costs. A thorough discussion about this can be found in [Briones et al., 2012, Kristoffersen et al., 2011], but one of the
major factors here is battery degradation. First, the impact of V2G on batteries lifetime is not fully understood yet. Second, vehicles and battery producers are not the final beneficiaries of V2G services. How is the warranty price to be changed when the battery is subject to such additional wear? From [Briones et al., 2012]: “EV OEMs are generally unwilling to permit discharge of energy from the battery by any control other than the vehicle’s powertrain control system. This assures that battery function is always tied to incremental miles added to the odometer, which contributes toward their warranty liability reduction. It also assures control over onboard components that need to meet high-reliability performance standards. By opening up the ESS to external control interfaces (i.e., price or regulation signals from utilities or aggregators), there is incurrence of additional risk. At this point in the design life cycle of the EV, as well as the market maturity of the industry, there is little perceived reward that offsets this risk. OEMs will have additional difficulty with this because they cannot know prior to vehicle delivery whether the owner will want to participate in V2G operations or not. Consequently, they will need to either provide all vehicles with the capabilities of long vehicle service with V2G operations or base warranty on other metrics such as the number of battery cycles”.

5.3.1 Model

Reflecting this discussion, we assume here that V2G power must be explicitly requested by the aggregator, otherwise it should not be activated. It is in particular not to be used to achieve better reference tracking. We indicate a V2G power request at time \( k \) with \( P^{\text{v2g}}[k] \), and with \( I^{\text{v2g}} \subseteq I \) we indicate the subset of EVs to be used to provide such service.

It is up to the aggregator to decide when it is opportune to make such \( P^{\text{v2g}} \) requests; such decision will depend on the payment the aggregator promises to the EV owners to
Figure 5.3: Local performance. Depicted are the initial state of charge, the desired final level, the battery capacity limits and the charge profile obtained using our scheme. All the local charging requirements are satisfied.

have the functionality available and the profit it retrieves from operating these services.

The model considered is then as follows.

- **Optimization Variables.** We introduce the new variable $v[i][k] \in \{0, 1\}$ to indicate a request sent to the $i$-th EV to inject power.

- **Objective Function.** The objective chosen is to

$$
\min_{u[i], v[i]} \sum_{k=0}^{N-1} \sum_{i \in I} P_i u[i][k] - (P^{\text{ref}}[k] + P^{\text{v2g}}[k]) + \gamma \sum_{i \in I} v[i][k],
$$

where $P^{\text{v2g}}[k]$ may or may not be added to the reference tracking objective, depending on the application. We rewrite the objective as

$$
\min_{u[i], v[i], r} \sum_{k=0}^{N-1} r[k] + \gamma \sum_{i \in I} v[i][k]. \tag{5.13}
$$

together with the constraint

$$
-r[k] \leq \sum_{i \in I} P_i u[i][k] - (P^{\text{ref}}[k] + P^{\text{v2g}}[k]) \leq r[k], \quad k \in \mathbb{N}_{[0, N-1]}.
$$
Figure 5.4: Fulfilment of the critical line capacity.

Figure 5.5: Primal and dual convergence.

Note that the effect of V2G power injections is not included in the tracking part of the objective directly, but by substituting $P^\text{ref}$ with $P^\text{ref} + P^{\text{V2G}}$. If the variables $v_i[k]$ were used to measure the tracking performance, then their use would have to be inhibited by a large enough $\gamma$. This would lead to a poor formulation, since it would be unclear, a-priori, what a sufficient magnitude for $\gamma$ is. In the proposed alternative, any $\gamma > 0$ ensures that $v_i[k]$ is used only for the fulfilment of constraint (5.14).

- **Coupling Constraints.** V2G power request is modelled as a constraint forcing a subset of the EVs $I^{\text{V2G}} \subseteq I$ to inject power in the network for a certain time:

$$
\sum_{i \in I^{\text{V2G}}} P_i v_i[k] \geq P^{\text{V2G}}[k] \quad k \in \mathbb{N}_{[0,N-1]},
$$

(5.14)

where $P^{\text{V2G}}[k]$ is the requested V2G power profile. As before, we also assume flow
constraints on some critical branch:
\[ \sum_{i \in \mathcal{I}} P_i u_i[k] \leq P_{\text{max}}[k] \quad k \in \mathbb{N}_{[0,N-1]}. \]

- **Subsystems Model.** The local model is updated to

\[
\begin{align*}
    e_i[0] &= E_i^{\text{init}} \quad (5.15) \\
    e_i[k + 1] &= e_i[k] + P_i \Delta T \left( \zeta_i^u u_i[k] - \zeta_i^v v_i[k] \right) \quad k \in \mathbb{N}_{[0,N-1]} \quad (5.16) \\
    e_i[N] &= E_i^{\text{ref}} \quad (5.17) \\
    E_i^{\text{min}} &\leq e_i[k] \leq E_i^{\text{max}} \quad k \in \mathbb{N}_{[0,N]} \quad (5.18) \\
    u_i[k] + v_i[k] &\leq 1 \quad k \in \mathbb{N}_{[0,N-1]} \quad (5.19) \\
    u_i, v_i &\in \{0,1\}^N, \quad (5.20)
\end{align*}
\]

where \( \zeta_i^u \approx 1 - \zeta_i \) is the charging conversion efficiency, while the discharging efficiency \( \zeta_i^v \) is \( \zeta_i^v \approx 1 + \zeta_i \). Condition (5.19) removes the possibility of charging and discharging simultaneously.

### 5.3.2 Solution Method and Results

We run the contraction solution scheme on this model. The procedure is the same as described in Section 5.2.1. The main difference with respect to the sole charging case is that now the optimal local solution is not greedy any more. It must be therefore solved either as a generic optimization problem, or by applying the Dynamic Programming (DP) algorithm, see e.g [Bertsekas, 2005, p.23]. In our tests we apply DP.

Figure 5.6 shows the simulation results for an instance with \(|\mathcal{I}| = 5000\) controlled EVs. Reference tracking performance, and the resulting “valley fill”, are depicted. Figure 5.7 shows the response of the EVs to the V2G power request, and the line constraint satisfaction on the critical line. Figure 5.8 shows the local charging behaviour of one EV; depicted are the local constraints, as well as the charge and discharge controls over the horizon.

Table 5.3 reports the computational results. The discharging functionality introduces a much more complicated combinatorial subproblem structure, since the local optimal strategy is not greedy any more. Models that exacerbate this, for instance when the objective is changed so that discharging is used to achieve price arbitrage, lead to instances that CPLEX cannot even finish preprocessing. Our proposed method has the

---

[1] The discharging efficiency must be greater than 1. This correctly encodes the fact that the amount of energy fed back to the network is smaller than the battery’s energy content decrease.
advantage of providing consistent solution times across different instances, and the solve times substantially outperform CPLEX. It should be emphasized that the computations are carried out on a single processor, so that solve times can be reduced substantially by exploiting parallelism.

Figure 5.6: Simulation results of an experiment with $|I| = 5000$ EVs. The power profile to be tracked, shown in (a), is designed to achieve a “valley fill” of the aggregated power consumption, as shown in (b).

(a) reference tracking  
(b) valley fill

Figure 5.7: Coupling constraints satisfaction: shown are (a) the aggregate response to the V2G power request, and (b) the satisfaction of line constraints on the critical branch.

(a) V2G request response  
(b) critical branch flow
Figure 5.8: The charge profile and control for one of the EVs. The upper picture illustrates the constraints included in the model: maximum and minimum energy levels (black), the desired final state of charge and the energy profile through the horizon. The bottom plot depicts the charge and discharge controls.

<table>
<thead>
<tr>
<th># PEVs</th>
<th>Proposed Method Tracking Error (%)</th>
<th>CPLEX Solve time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tracking Error (%)</td>
<td>Solve time (min)</td>
</tr>
<tr>
<td></td>
<td>Min</td>
<td>Avg</td>
</tr>
<tr>
<td>50</td>
<td>3.89</td>
<td>5.29</td>
</tr>
<tr>
<td>100</td>
<td>2.52</td>
<td>3.46</td>
</tr>
<tr>
<td>200</td>
<td>1.59</td>
<td>2.13</td>
</tr>
<tr>
<td>500</td>
<td>0.89</td>
<td>1.15</td>
</tr>
<tr>
<td>1000</td>
<td>0.55</td>
<td>0.70</td>
</tr>
<tr>
<td>2000</td>
<td>0.40</td>
<td>0.46</td>
</tr>
<tr>
<td>5000</td>
<td>0.23</td>
<td>0.25</td>
</tr>
<tr>
<td>10000</td>
<td>0.13</td>
<td>0.18</td>
</tr>
</tbody>
</table>

Table 5.3: Charging and V2G.
6. A Solution Method based on Primal Averaging

In the previous chapters we have always worked under the assumption that an exact optimal dual solution $\lambda^*$ was available. In this chapter we are interested in the behavior of $x(\lambda)$ when $\lambda^{[k]} \rightarrow \lambda^*$, and we will exploit the fact that it is possible to recover a solution to $P_{LP}$, denoted by $x_{LP}^*$, during the process of solving $D$.

To achieve these results, we borrow a method from convex optimization which was initially proposed in [Shor et al., 1985]. We will outline this method in the next section. For the moment, notice that a solution to the convexication $P_{LP}$ may, in general, not necessarily satisfy the local constraints (in particular, the integrality conditions on the discrete variables). However, in the following theorem we establish that, for the specific structure of $P$, a vertex solution of $P_{LP}$ is indeed useful as it may violate the local constraints only for a few subsystems.

**Theorem 6.1.** Let $\bar{x}$ be a vertex of $P_{LP}$. Then $\bar{x}_i \in \text{conv}(X_i)$ for all $i \in I$, and there exists $I_1 \subseteq I$, with cardinality at least $|I_1| \geq |I| - m - 1$, so that $\bar{x}_i \in X_i$ for all $i \in I_1$.

**Proof.** See Appendix C.7. \qed

Note that Theorem 6.1 holds for any vertex of the feasible set of $P_{LP}$, and thus in particular also for any vertex optimizer $x_{LP}^*$.

**Remark 6.1.** This result complements Theorem 3.2, which concerns the properties of the inner solutions $x(\lambda^*)$ and requires Assumption 3.1 to work (see counterexample in Appendix A.4). Theorem 6.1 is about the structural properties of any vertex of the feasible set of $P_{LP}$, and holds without any assumption on uniqueness of primal or dual optimizers. Notice that the result given in Theorem 6.1 is slightly weaker, because the number of problematic subsystems is $m + 1$ (vs. $m$ given in Theorem 3.2). This is due to the technique used to prove the result: here we made direct use of the Shapley–Folkman–Starr Theorem C.1, while to prove the stronger version in Theorem 3.2 we used an argument based on simplex tableaus. It is possible to apply the same argument to prove Theorem 6.1, and hence get the tighter result, but in the thesis we rather wanted to include both proof strategies.
In the next section we show how to recover $x^*_L$, while in Section 4 we exploit it in order to recover a good feasible solution for the original $P$.

### 6.1 Averaging Method

In this section we discuss a method to obtain a solution $x^*_L$ to $P_L$. It is based on a method developed for convex problems, which was first proposed in [Shor et al., 1985], was then further developed in [Larsson and Liu, 1997] and has been recently extended by [Nedic and Ozdaglar, 2009]. Its theoretical properties in the convex case are summarized in [Anstreicher and Wolsey, 2009]. It consists in deploying a combination of the subgradient method and an averaging scheme for the inner solutions.

The subgradient method is a well known method used to solve duals. According to it, given an initialization $\lambda^{[1]}$, the subsequent dual iterates $\lambda^{[k]}$ are updated as

$$\lambda^{[k+1]} = \Pi_+(\lambda^{[k]} + s^{[k]} \cdot \gamma^{[k]}),$$

in which $\Pi_+(\cdot)$ denotes the projection onto $\mathbb{R}_m^+$, $s^{[k]}$ is the step length and $\gamma^{[k]}$ is a subgradient of the dual function. When solving a dual problem, a valid subgradient is given by the residuals of the current inner solution, i.e., $\gamma^{[k]} = \sum_{i \in I} H_i x_i^{[k]} - b$.

Following the scheme proposed in [Anstreicher and Wolsey, 2009], we use the steplength rule

$$s^{[k]} = \frac{\alpha}{k},$$

and as we produce the dual iterates $\lambda^{[k]}$ during the subgradient optimization (6.1), we construct an average of the inner solutions encountered:

$$\bar{x}^{[k]} = \frac{1}{k} \sum_{j=1}^{k} x(\lambda^{[j]}).$$

Surprisingly\(^1\), the sequence $\{\bar{x}^{[k]}\}$ accumulates at the optimizers of $P_L$ [Anstreicher and Wolsey, 2009, Cor. 5]. For the sake of simplicity, we now make the following assumption.

**Assumption 6.1.** The optimization problem $P_L$ has the unique solution $x^*_L$.

Assumption 6.1 eliminates the need to talk about accumulation points, enabling the following concise convergence result.

\(^1\)Recall that, as shown in Example A.1, the iterates $x(\lambda^{[k]})$ are generally non-unique, suboptimal, or even infeasible points.
Theorem 6.2 (primal and dual convergence). Suppose that the subgradient method (6.1) is applied to the dual problem $\mathcal{D}$. Then $\hat{\lambda}^{[k]} \to \lambda^* \in \Lambda^*$, where $\Lambda^*$ is the set of optimal dual solutions. Further, if Assumption 6.1 holds, then $\hat{x}^{[k]} \to x^*_\mathcal{P}$.

Proof. Dual convergence $\lambda^{[k]} \to \lambda^* \in \Lambda^*$ is proven in [Anstreicher and Wolsey, 2009, Thm. 2]. For primal convergence, see Appendix C.8. □

Remark 6.2. Dual convergence asserted in Theorem 6.2 can be generalized to any step length rule which satisfies

$$s^{[k]} \to 0, \quad \sum_{k=1}^{\infty} s^{[k]} = \infty, \quad \sum_{k=1}^{\infty} (s^{[k]})^2 < \infty,$$

(6.3)

see [Anstreicher and Wolsey, 2009, Thm. 3]. Other averaging schemes that generalize (6.2) can be found in [Sherali and Choi, 1996].

6.2 Application of the Averaging Scheme to $\mathcal{P}$

In this section we propose an algorithm to recover a good solution to $\mathcal{P}$. We take advantage of the properties of vertex solutions of $\mathcal{P}_\mathcal{LP}$ established in Theorem 6.1, and the primal solution recovery scheme of Theorem 6.2. In order to simplify the analysis and the presentation, we work under the following assumption.

Assumption 6.2. The local systems are such that, for all $i \in I$, $0 \in X_i$ and $H_i x_i \geq 0$ for all $x_i \in X_i$.

We discussed this assumption back in Chapter 4.2 when introducing conservatism reduction mechanisms. Here, it allows us to devise a simple scheme to rectify $x^*_\mathcal{LP}$ without contraction. The assumption essentially requires that any local choice made “consumes” budget ($H_i x_i \geq 0$), and that “not picking anything” is possible ($0 \in X_i$). It is true for a number of traditional combinatorial problems (e.g. the knapsack problem) as well as more refined models used in practice, such as portfolio optimization problems for small investors, for which models fitting $\mathcal{P}$ have been proposed [Baumann and Trautmann, 2013]. Another example application is the supply chain problem presented in Chapter 5.

On the other hand, this assumption would not hold for the EV charging coordination problem: at some point, EVs do need to be charged (hence $0 \notin X_i$), and in the configuration when EVs can also inject power back in the network, such operations contribute to increase the power line budget.

Our proposed method to solve $\mathcal{P}$ under Assumptions 6.1, and 6.2 is reported in Algorithm 6.1. It is a two–phases method.
Algorithm 6.1 Primal Solution Recovery

Initialization:

\[ k = 1 \]
\[ \lambda^{[1]} = 0 \]

Construction of the Ergodic Sequence:

\begin{algorithmic}
  \While {\( k \leq k_{\text{max}} \)}
    \State \( x_i(\lambda^{[k]}) = \arg \min_{x_i \in X_i} (c_i + \lambda^{[k]} H_i) x_i \)
    \State \( \bar{x}_i^{[k]} = \frac{1}{k} \sum_{j=1}^{k} x_i(\lambda^{[j]}) \)
    \State \( \gamma^{[k]} = \sum_{i \in I} H_i x_i(\lambda^{[k]}) - b \)
    \State \( \lambda^{[k+1]} = \max(\lambda^{[k]} + \frac{\alpha}{k} \gamma^{[k]}, 0) \)
    \State \( k = k + 1 \)
  \EndWhile

Rectification of the Solution:

\begin{algorithmic}
  \State determine the partitioning of \( I = I_1 \cup I_2 \), such that for all \( i \in I_1 \), \( \bar{x}_i^{[k_{\text{max}}]} \in X_i \) (in particular, it satisfies integrality). \(|I_1| \geq |I| - m - 1\).
  \For {\( i \in I_1 \)}
    \State \( \hat{x}_i^* = \bar{x}_i^{[k_{\text{max}}]} \)
  \EndFor
  \State \( \rho = \sum_{i \in I_1} H_i \hat{x}_i^* \)
  \State \( (\hat{x}_i^*)_{i \in I_2} = \begin{cases} 
    \arg \min_{i \in I_2} \sum_{i \in I_2} c_i x_i \\
    \text{s.t.} \quad \sum_{i \in I_2} H_i x_i \leq b - \rho \\
    x_i \in X_i, i \in I_2
  \end{cases} \quad (P_{\text{FIX}}) \)
\end{algorithmic}

In a first phase, using averaging we construct \( x_{i \text{p}}^* \) and detect \( I_1 \), the subset of subsystems whose solution already satisfies all the local constraints (including the integrality requirements). We let the averaging run for \( k_{\text{max}} \) iterations, where \( k_{\text{max}} \) is a parameter chosen such that the averaging sequence has settled. Recall that the convergence of this sequence is guaranteed by Theorem 6.2.

In the second phase, we reoptimize over the smaller set of subsystems indexed by \( I_2 = I \setminus I_1 \). The second optimization is low dimensional since it is guaranteed to entail at most \( m + 1 \) subsystems. The input of the algorithm is the data of the problem, i.e. the tuple \((c_i, H_i, A_i, d_i)_{i \in I}\) for each subsystem, and the resource vector \( b \in \mathbb{R}^m \). The output is \( \hat{x}^* \), a feasible solution to \( P \) that satisfies the performance bound given in the following Theorem.
Theorem 6.3. Under Assumptions 6.1 and 6.2, the solution $\hat{x}^*$ produced by Algorithm 6.1 is a feasible solution to $\mathcal{P}$ which satisfies the following performance bound:

$$J_\mathcal{P}(\hat{x}^*) - J_\mathcal{P}^* \leq (m + 1)\max_{i \in I} \max_{x_i \in X_i} c_i^T x_i.$$  \hspace{1cm} (6.4)

Proof. See Appendix C.9. \hfill \Box

According to the performance bound (6.4), if $J_\mathcal{P}$ grows linearly as we increase the size of the problem $|I|$, and if the sets $\{X_i\}_{i \in I}$ are uniformly bounded, then

$$\frac{J(\hat{x}^*_i) - J_\mathcal{P}^*}{J_\mathcal{P}} \to 0 \quad \text{as} \quad |I| \to \infty,$$  \hspace{1cm} (6.5)

indicating that the quality of the solutions recovered improves as the problem size is increased.

Remark 6.3. The reoptimization in $\mathcal{P}_{\text{fix}}$ is not necessary, one can safely pick $\hat{x}^*_i = 0$ for $i \in I_2$. The advantage of opting for this variation is that then the algorithm is fully distributed. And while the performance of the solutions is decreased in this case, it still satisfies the bound (6.4).
7. Application – Supply Chains

The main objective of this section is to assess how the method proposed in Chapter 6 performs on a mixed integer optimization problem that is of practical interest: the problem of partial shipments.

In this problem setting, a distributor of some products has to supply multiple customers. Due to uncertainties in the demand and high storage costs, as well as restrictions on manufacturing capacities (especially true for seasonal items such as ski equipment or pharmaceuticals), the products available in the inventory that are ready for shipping is less than the total demand. Under these circumstances, more often than not, the distributor chooses to satisfy partial orders of more customers, instead of fully satisfying only a few of them [Dawande et al., 2006]. On the other hand, these shipments cannot be too small, due to transportation costs and the additional paperwork and tracking costs. The problem distributors face is thus to allocate the product inventories available to customers in the presence of shipping restrictions.

7.1 Model for the Optimization Problem

We use the formulation proposed in [Dawande et al., 2006], which we summarize here for completeness. We are given a demand of $M$ products from $N$ customers. $D_j^i$ is the demand of product $j$ from customer $i$, while $I_j$ is the amount of product $j$ available in the inventory. If a shipment is made, it must at least amount to the $\beta$-fraction of the total demand over all products. The optimization variables are $w^i \in \{0, 1\}$, which decide whether customer $i$ gets a partial shipment, and $S_j^i$, which is the amount of product $j$ shipped to customer $i$.

The rewards are composed of two components. A fixed reward amounting to $K^i$ is obtained if a shipment is made to customer $i$. This can encode the customer appreciation for receiving shipment, even if partial, or the weight that the distributor attaches to each customer. There is an additional revenue $r_j^i$ that the supplier retrieves based on what fraction $S_j^i$ of the total demand $D_j^i$ is shipped.
We can formulate this problem as the following optimization program

\[
\begin{align*}
\text{minimize} & \quad - \sum_{i \in I} K^i w^i - \sum_{i \in I} \sum_{j \in J} r_j^i \frac{S_j^i}{D_j^i} \\
\text{subject to} & \quad \sum_{i \in I} S_j^i \leq I_j, \quad j \in J \\
& \quad \sum_{j \in J} S_j^i \geq \beta \cdot w^i \sum_{j \in J} D_j^i, \quad i \in I \\
& \quad 0 \leq S_j^i \leq w^i \cdot D_j^i, \quad i \in I, j \in J \\
& \quad w^i \in \{0, 1\}, \quad i \in I.
\end{align*}
\] (7.1)

**7.2 Applying the Method based on Averaging**

The partial shipment problem (7.1) is NP-hard, which is proven by reduction to a knapsack problem, see [Dawande et al., 2006, Lemma 2.1]. It is also known that greedy strategies perform poorly on it [Dawande et al., 2006]. We tried to solve it using general purpose solvers, but these are also inadequate for the larger instances (see Table 7.2). In [Dawande et al., 2006], the authors propose a heuristic to obtain good partial assignments that can be fixed, thus reducing the number of integer variables to be solved. The method used to retrieve the assignment is relatively sophisticated and must be adapted when the model is subject to changes. Even under these circumstances, solving the problem remains a challenge: reported solve times for medium sized instances are up to 6 hours, with an average optimality gap of the recovered solutions of 6.2%.

We tackle the problem using our proposed method. Here duality can be exploited by relaxing the budget constraint (7.1b). This leads to the following dual problem

\[
\begin{align*}
\max_{\lambda \geq 0} \min_{w, S_j^i} & \quad - \sum_{i \in I} K^i w^i - \sum_{i \in I} \sum_{j \in J} \left( \frac{r_j^i}{D_j^i} - \lambda_j \right) S_j^i - \sum_{j \in J} \lambda_j I_j \\
& \quad \sum_{j \in J} S_j^i \geq \beta \cdot w^i \sum_{j \in J} D_j^i \\
& \quad 0 \leq S_j^i \leq w^i \cdot D_j^i \\
& \quad w^i \in \{0, 1\},
\end{align*}
\] (7.2)

whose inner problem is decoupled in \( N \) separate optimization programs, one for each customer. We apply Algorithm 6.1 using this relaxation. For the tests, we generate 10 instances of the problem using random parameters sampled from a uniform distribution over the following ranges: \( D_j^i = U[1 - 100] \), \( K^i = U[1 - 100] \), \( r_j^i = U[1 - 15] \), \( \beta = 0.6 \).
The budgets $I_j$ are assigned according to Table 7.1. These parameter values are taken from [Dawande et al., 2006], and the sizes of the problems considered are representative for industrial instances related to a pharmaceutical company. The computations are carried on a Laptop PC with 4GB of RAM and a 2.67GHz processor.

The outer (maximization) problem in (7.2) is solved using a subgradient method. Before activating the step length rule $\alpha/k$, we first perform 50 iterations with fixed step length, in order to localize a better initialization point $\lambda^k$ for Algorithm 6.1.

At each iteration, the inner (minimization) problem in (7.2) is solved as a generic optimization program. As opposed to the fully coupled system (7.1), general purpose solvers solve the decoupled problem very rapidly. The inner problems are constructed using Yalmip [Löfberg, 2004] and solved by CPLEX 12.5.

Figure 7.1(a) depicts the typical dual convergence observed during the execution of Algorithm 6.1. The vertical dashed line indicates the iteration from which we activate the $\alpha/k$ stepsize rule and start to compute the average sequence. Figure 7.1(b) shows primal convergence of the averaged $\tilde{x}^k$. In our experiments the “heavy tail” behavior shown is dominated by the continuous part of the problem, i.e., by the convergence rate of $(\tilde{x}^k)^i \rightarrow (S^*_{iP})^i$. In contrast, the convergence of the integer part, i.e., $(\tilde{w}^k)^i \rightarrow (w^*_{iP})^i$ is much faster. In order to speed up the computations, we thus fix the integer parts of $(w^k_{I_{max}})^i$, and in the rectification part of the Algorithm, where we recompute a solution over $I_2$ using the remaining budget, we also recompute the continuous part of the problem $(S^j)$ for all the subsystems.

The results are reported in Table 7.2, where we compare the performance of our proposed method with CPLEX 12.5 when it is applied to the fully coupled system (7.1). CPLEX is generally unable to provide exact solutions to the problem before running out of memory. Our proposed method provides nearly optimal feasible solutions, the computation times are acceptably short ($\leq 5$ min) and are affected by a low degree of variance. We tested the algorithm on a single processor, but since the bulk of the computational effort lies in the computation of solutions to the inner problem, solve times can be substantially improved by exploiting parallelism.
<table>
<thead>
<tr>
<th>N</th>
<th>M</th>
<th>Proposed Method</th>
<th>CPLEX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Gap (%)(^\dagger)</td>
<td>Time (s)(^\dagger)</td>
</tr>
<tr>
<td>100</td>
<td>25</td>
<td>.11-.38-.60</td>
<td>16.7-18.5-21.5</td>
</tr>
<tr>
<td>300</td>
<td>75</td>
<td>.01-.03-.07</td>
<td>61.0-62.5-63.4</td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>.04-.06-.11</td>
<td>95.0-101.0-110.6</td>
</tr>
<tr>
<td>600</td>
<td>50</td>
<td>.01-.06-.10</td>
<td>113.7-116.6-119.5</td>
</tr>
<tr>
<td>1000</td>
<td>100</td>
<td>.00-.03-.05</td>
<td>225.3-245.2-288.8</td>
</tr>
</tbody>
</table>

\(^\dagger\)indicated are min., average and max. computation times.

(-) runs out of memory before solving.

Table 7.2: Performance of the proposed method compared to CPLEX 12.5.

Figure 7.1: Primal and dual convergence.
Part II

Convex Approximations based on Linear Relaxations, for the Robust Optimization of Uncertain Systems
8. Introduction to the Part and Outline

The main topic of this part is the computation of robust solutions to uncertain models entailing integer variables. Linear relaxation schemes will play a role in making the computations of these solutions faster.

Generally speaking, the task of Robust Optimization is to take the known data of an uncertain problem, together with information available on the uncertainty (for instance, bounds on the values uncertain terms may attain) and return a so-called Robust Counterpart (RC). The RC is another optimization problem and, in contrast to the initial uncertain program, it is deterministic (all the data is known when it is to be solved). It is constructed in such a way that its solutions remain feasible for the initial uncertain problem for any possible realization of the uncertainty.

Known Results. The first work introducing these ideas appears to be [Soyster, 1973], while the main technique of robust optimization, in which an argument based on duality is used to replace the universality quantifier $\forall$ with an existence one $\exists$, was first proposed in [Thuente, 1980]. This technique has been crucial for many subsequent approaches, including some of the most recent ones.

The robust optimization framework is most useful when the RC can be formulated as a finite optimization problem, in which case it is also sometimes referred to as the explicit RC. Nowadays, explicit RCs exist for a number of practically important cases. The recent book [Ben-Tal et al., 2009] contains many of these results.

These RCs are also immediately applicable to models entailing integer variables, as noted in [Ben-Tal et al., 2009, Remark 1.2., p.26]. In this case, however, the computational tractability of the resulting RC is under question, as the original nominal model may not be tractable itself. And even when the nominal problem is tractable, adding variables and constraints in the robustification procedure may destroy the underlying advantageous structure. One notable exception to this is in [Bertsimas and Sim, 2003].

Much of recent research effort in this area has been directed towards making robust optimization approaches less conservative.

One direction has been to look for new uncertainty models. For instance, in [Bertsimas and Sim, 2004], the authors investigate a setting in which the amount of uncertainty is
“budgeted”. Conservatism is reduced by assuming that, while all the data may be affected by uncertainty, within a certain realization only a subset of the data is actually changed from its nominal value. The size of this subset is then the budget.

A second line of research has been directed towards approaches that react to uncertainty as it unveils, in the context of multistage decision problems. An optimal recourse policy in this setting can be obtained using Dynamic Programming, which, however, becomes computationally intractable even for modest model sizes. This difficulty can be overcome by restricting the class of policies considered in the optimization, in particular to affine policies [Ben-Tal et al., 2004a, Goulart et al., 2006]. In this case a trade-off is made between performance and computational tractability.

In the following chapters we look further into both of these research directions, while insisting on robustification methods that are of little added computational burden.

Outline. In Chapter 9 we consider generic mixed integer problems and a new model for the uncertainty set, which is now a function of the decision taken.

We show how to compute a robust solution in this case and how this framework is useful for practical purposes, for instance to optimize schedules subject to uncertainty. As a practical application example, we study the problem of reserve provision for industrial electricity consumers.

The robust counterpart developed applies to integer as well as mixed-integer linear programs. It is furthermore computationally advantageous because the problem class is retained, the number of discrete variables is the same as the deterministic nominal problem, and the new model’s data necessary to determine the robust solution is obtained through extremely lightweight computations.

In Chapter 10 we discuss Model Predictive Control of systems with mixed discrete and continuous inputs. These usually require the online solution of a mixed integer optimization problem. Here we propose an approximation approach in which the integer input constraints are initially relaxed to an interval, so that the resulting optimization problem to be solved online is a convex quadratic program (QP). This is computationally much less demanding than a mixed integer problem, both in theory and practice. A projection is then applied to the relaxed solution in order to obtain inputs satisfying the integer constraints.

The novelty of the proposed approach is the integration of the projection operation within a robust optimization framework. Input constraints are satisfied by design of the projection function, while the robustification procedure (constraint tightening) guarantees state constraints satisfaction for the projected input sequence. For this we introduce an artificial uncertainty in the model, which encodes the maximum possible discrepancy between the state evolution under the projected and the relaxed control sequences. The corre-
sponding constraint tightening can be precomputed, and hence the method is applicable with no additional online computational effort.

We find the class of Pulse-Width Modulated (PWM) systems particularly well suited for this method. We present a suitable projection function for these systems, and with experiments on different power converter topologies we show one can obtain substantially enhanced control performances using our method with respect to a controller based on plain linear MPC – at the same computational effort. The choice of the projection function is the key for good performance.

In Chapter 11 we examine an extension of the previous Chapter in which we want to use the possibility of recourse to diminish the amount of conservatism introduced by the robustification procedure.

For linear systems with bounded disturbances, optimization over control laws that are affine in the disturbance measurements has emerged as a valid trade-off between performance and computational expense. In this chapter, we extend the idea of affine recourse to hybrid systems, and introduce an approach that chooses integer decision variables over the control horizon in conjunction with continuous decisions that are subject to recourse.

The efficacy of the proposed approach is illustrated on a DC-DC buck power converter, and compared to an implementation of affine control policies based on a linearized model.
9. Robust Mixed–Integer Optimization, with Application to Uncertain Scheduling Problems

In the first part of this Chapter we explain what nominal and uncertain models we consider. These do not fit traditional robust optimization frameworks, so we derive a new Robust Counterpart (RC). The computations required to determine our proposed RC are typically lightweight.

In the second part we examine how these results are of practical interest in the context of scheduling problems under uncertainty, with a particular application to reserve provision for industrial electricity loads. We illustrate how the method is to be applied on an example involving the scheduling of two cement milling machines.

9.1 Optimization Problems Considered and their Uncertain Version

In this chapter we consider the generic integer optimization problem

\[
(IP) : \begin{cases} 
\min & c^T x \\
\text{subject to} & Ax \leq b \\
& x \in \{0, 1\}^n 
\end{cases}
\]

and the following uncertain version associated to it\(^1\)

\[
(UIP) : \begin{cases} 
\min & c^T x \\
\text{subject to} & Ax + Dw \leq b \\
& w \in W(x) \\
& x \in \{0, 1\}^n 
\end{cases}
\]

\(^1\)This modeling also includes the case in which uncertainty enters in the objective as an additive disturbance \(d^T w\). In such a case, one introduces a new variable \(t \geq c^T x + d^T w\) and then sets it as the minimization objective function.
where \( A \in \mathbb{R}^{m \times n} \), \( b \in \mathbb{R}^m \), \( c \in \mathbb{R}^n \) and \( D \in \mathbb{R}^{m \times p} \). Problem \((UIP)\) is uncertain in the sense that the realization of the term \( w \in W(x) \) is unknown at the time when the problem is to be solved. Here we want to deal with this uncertainty in a robust fashion, i.e., we would like to find a solution \( x^* \) that remains feasible for the constraints in \((UIP)\) for any possible realization of \( w \in W(x) \) and, in particular, among all candidates that satisfy this property, we would like to obtain one achieving the best possible objective.

In contrast with other literature on robust optimization [Ben-Tal et al., 2009, Bertsimas et al., 2011], here we assume that the uncertainty set \( W \) is not a static set, but rather a function of the decision \( x \), i.e., a set-valued function:

\[
W : \mathbb{R}^n \to \mathbb{R}^p
\]

\[
x \mapsto W(x).
\] (9.1)

Models of uncertainty that depends on the decision taken have already appeared in the literature, in the context of stochastic optimization and applied to the planning of offshore oil or gas field infrastructure [Goel and Grossmann, 2006, Tarhan et al., 2009]. Note that the robust counterpart (RC) to this problem, in the general case where \( W \) is any arbitrary set-valued map, leads to very difficult (or even impossible to solve) optimization problems. The particular type of dependence studied here is

\[
W(x) = \bigoplus_{k=1}^n W_k(x[k]) = \bigoplus_{k=1}^n x[k] \cdot W_k.
\] (9.2)

in which \( \oplus \) indicates the Minkowski sum of sets, \( W_k \) is a compact set of vectors in \( \mathbb{R}^p \), and the multiplication means the product of each element in \( W_k \) by the scalar quantity \( x[k] \).

In Section 9.3 we will give an example of how this structure may be of practical interest in the context of scheduling under uncertainty. In the next section we show that it is possible to compute a robust solution to \((UIP)\) with uncertainty structured as in \((9.2)\) by solving a problem that is not harder than \((IP)\).

### 9.2 The Explicit Robust Counterpart

In the following Theorem we determine how to compute a robust (or worst-case) solution to \((UIP)\) by providing its explicit Robust Counterpart.

**Theorem 9.1.** The explicit Robust Counterpart to \((UIP)\) is the following deterministic optimization problem

\[
(RC - UIP) : \begin{cases}
\min & c^T x \\
\text{subject to} & (A + H)x \leq b \\
& x \in \{0,1\}^n
\end{cases}
\]
in which the $i$-th row, $k$-th column entry of the matrix $H$ is

$$H_{ik} = \sigma_{\text{conv}(W_k)}(d_i) = \max_{w \in \text{conv}(W_k)} d_i \cdot w,$$

(9.3) where $\sigma_{\text{conv}(W_k)}$ is the support function of $W_k$, and $d_i$ is the $i$-th row of the matrix $D$.

Proof. See Appendix C.10.

Note that the computation in (9.3), which determines the matrix $H$, can be carried out before solving $(RC - UIP)$ since it does not depend on $x$. Furthermore, the maximization can be done over $\text{conv}(W_k)$ or $W_k$, whichever is easier. Intuitively, this follows from the fact that the worst case scenario happens at a vertex of the uncertainty set. In the scheduling example, we have access to this convex hull, and so in principle we could solve (9.3) as a (low dimensional) linear program. However we will see we don’t even need to trouble ourselves with this, since in that case we have an analytic solution for (9.3).

Note further that $(RC - UIP)$ does not involve the solution of a larger integer optimization problem. This is one crucial feature of the method. In order to obtain a robust solution, the only additional burden is the computation of the matrix $H$; then, an optimization problem of the exact same size as the original nominal problem has to be solved, in which the only difference is that its data is changed.

Finally, it is possible to extend this result to generic mixed integer problems; however, when the variables on which $W(x)$ depends can attain negative values, and $W_k$ is not symmetric around the origin, we need to make a slight adjustment to the uncertainty model and the explicit Robust Counterpart, as we indicate in the next theorem.

**Theorem 9.2.** Consider the following uncertain mixed-integer optimization problem

$$(UMILP): \begin{cases} \min & c^T x \\ \text{subject to} & Ax + Dw \leq b \quad w \in W(x) \\ & x[k] \in \mathbb{Z} \quad k \in K_z \end{cases}$$

where $K_z$ is the index set of integer components of $x$, and in which uncertainty is structured as

$$W(x) = \bigoplus_{k=1}^n W_k(x[k]) = \bigoplus_{k=1}^n |x[k]| \cdot W_k.$$

Then, its explicit Robust Counterpart is

$$(RC - UMILP): \begin{cases} \min_x & c^T x \\ \text{subject to} & Ax + Hy \leq b \\ & -y \leq x \leq y \\ & y \geq 0 \\ & x[k] \in \mathbb{Z} \quad k \in K_z, \end{cases}$$

where $H$ is to be computed as indicated in (9.3).
9 Robust Mixed-Integer Optimization, with Application to Uncertain Scheduling Problems

Proof. See Appendix C.11.

In this case the size of the problem is larger because the variable \( y \) has to be introduced. However, \( y \) is continuous, meaning that in the worst case scenario, the maximum number of enumerations necessary to solve the problem (i.e., the size of the tree in a branch\&bound method) is the same as for the original nominal problem.

In the next section we discuss a possible application of these results.

9.3 Application – Electricity Cost Minimization and Reserve Provision of an Industrial Power Consumer

In an interconnected electricity network, the amount of power consumed must be equal to the power produced at any time, because uncompensated differences between inflows and outflows eventually lead to blackouts. Network operators are responsible for managing in real time the ancillary services that ensure this balancing, which can be done by controlling the production of privately owned generation units. These services are categorized according to their activation time, and they constitute the so-called network reserves.

In an environment in which the penetration of uncontrollable, intermittent renewable sources is growing at a fast pace (see Figure 9.1), ensuring this balance is maintained while providing electricity to consumers at any time and in any desired amount is an operational challenge, since network operators have to minimize the usage of expensive reserves while ensuring a functioning network. One way to reduce the ensuing increase in required reserves is to make consumers sensitive to the current generation situation, implying that consumption can be controlled in exchange for economic benefits. This is generally known as demand side management (DSM), and can be seen as a way to integrate consumers in the aforementioned network reserves.

Here we study the problem of incorporating large electricity consumers in such reserve mechanisms. The economic potential of this incorporation has been widely recognized [Associates, 2005, NERC, 2007, ALCOA, 2009] and indeed many large consumers, in particular large producers of chloride, aluminium and steel, are already marketing part of their consumption as reserves to make profits [Paulus and Børggrefe, 2011].

Large consumers can provide reserves by shifting some of their consumption in time (load shifting), by decreasing it (load shedding) or both, depending on a mix of technical and economic constraints. Here we focus on the load shifting approach since, as will be discussed in Section 9.4.1, it is economically vastly more competitive than load shedding.
In practice, these will typically consist of shifts of 15–30 minutes around nominal schedules (as opposed to large shifts of processes from e.g. day to night) since the capacity utilization factors for energy-intensive industries are high (80% and above).

The challenge here is that whether a reserve request is made or not is not known at the time when the consumption schedule is decided. This is the uncertain feature in this context, and is interpreted as uncertainty affecting the start time of the tasks.

**Model.** Electricity cost minimization problems have been developed for many specific process industries, e.g. air separation plants [Mitra et al., 2011], continuous paper pulp production [Pulkkinen and Ritala, 2008] and cement plants [Castro et al., 2011] (and many more references therein). This has become an active area of research in recent years, as large consumers see an opportunity to cut daily operational expenses, for example by purchasing energy more directly from the new competitive electricity markets. The resulting optimization problems can range from simple linear optimization problems to more difficult large-size non-linear and discrete programs.

Here we consider discrete-time formulations which result in MILPs. Suppose we are given the task of scheduling the electricity consumption of an industrial load with the aim of minimizing the electricity costs. The problem is to schedule a given set of (energy intensive) tasks \( i = 1, \ldots, M \) within a discrete-time horizon indexed by \( k = 1, \ldots, N \). Suppose that the optimization variable in these problems is \( x_i \in \{0, 1\}^N \), the start time of job \( i \), i.e., \( x_i[k] = 1 \) is job \( i \) is to be started at time step \( k \); otherwise \( x_i[k] = 0 \).

According to (IP), we restrict our attention to scheduling problems in which the constraints on \( x_i \), as well as the objective, are exclusively linear. A prototypical model may be as follows.
Objective. The electricity cost minimization objective can be formulated as

\[ \min_{x} \sum_{i=1}^{M} \lambda^\top P_i x_i \]  

(9.4)

where for each task \( i \), the power-consumption matrix \( P_i \) is constructed as

\[
P_i = \begin{bmatrix}
p_i^1 & 0 & \cdots & 0 \\
p_i^2 & p_i^1 & \ddots & \\
\vdots & \ddots & \ddots & 0 \\
p_i^l & \vdots & p_i^1 & \cdots \\
0 & p_i^l & p_i^2 & \ddots \\
\vdots & \ddots & \ddots & \ddots \\
0 & \cdots & 0 & p_i^l
\end{bmatrix},
\]

(9.5)

in which \( l_i \) is the length (in time steps) of job \( i \). This matrix encodes the power profile resulting from starting job \( i \) at time step \( k \), since the product \( P_i x_i \) selects the column of \( P_i \) for which \( x_i[k] = 1 \). Note that a \( p_i = 0 \in \mathbb{R}^N \) profile may be chosen, in which case task \( i \) does not necessitate power, but it must still be considered in the scheduling.

The cost \( \lambda \in \mathbb{R}^N \) is the (deterministic) vector of electricity prices over the scheduling horizon. This profile is assumed to be known, which is a reasonable assumption if the industrial consumer purchases electricity on forward markets one week in advance, at the time when the schedule has to be decided.

Constraints. We assume that the scheduling is subject to the following set of constraints.

- We first require that each task is started exactly once within the scheduling horizon:

\[ 1^\top x_i = 1 \quad \forall i \]  

(9.6)

- We assume we have minimum and maximum total power consumption constraints for each time step period, which we can encode as

\[ p \leq \sum_{i=1}^{M} P_i x_i \leq \bar{p}. \]  

(9.7)

- We also assume we have sequentiality constraints on the jobs. To encode these, we introduce the matrix \( S \in \mathbb{R}^{M \times M} \): whenever the \( S_{ij} \) element is 1, job \( j \) has to come after job \( i \) (otherwise the entry is 0). The constraint can then be expressed as

\[ S_{ij} \cdot (h_{1} x_j) \geq S_{ij} \cdot (h_{1} x_i + l_i), \quad \forall (i,j) \]  

(9.8)

where \( h_1 = [1, 2, \ldots, M] \).
• Constraints on the earliest delivery and due times of job \( i \); these are determined by the constants \( d_i, \overline{d} \in \mathbb{R}^M \).

\[
\begin{align*}
  d_i & \leq h_1 x_i - \sum_{j \neq i} S_{ji}(h_1 x_j + l_j) \quad \forall i \\
  h_1 x_i - \sum_{j \neq i} S_{ji}(h_1 x_j + l_j) & \leq \overline{d}_i \quad \forall i
\end{align*}
\] (9.9) (9.10)

• We may additionally require that each job must be started early enough such that it is completed within the horizon, i.e.,

\[
h_2^i x_i = 0 \quad \forall i,
\] (9.11)

where \( h_2^i = [0, \ldots, 0, 1, \ldots, 1] \).

In summary, thus, the scheduling optimization model we will consider here is

\[
\begin{aligned}
\min & \quad \text{electricity costs (9.4)} \\
\text{subject to} & \quad \text{consistency (9.6)} \\
& \quad \text{min/max power consumption (9.7)} \\
& \quad \text{sequentiality (9.8)} \\
& \quad \text{earliest delivery/due time (9.9)} \\
& \quad \text{job termination (9.11)},
\end{aligned}
\] (9.12)

which fits the generic model \((IP)\), once we stack the decisions \( x_i \) for each task \( i \) into the vector \( x = [x_1, \ldots, x_I] \in \mathbb{R}^{N \times M} \). In Section 9.4 we will look at a specific application where the model (9.12) can be used; it involves the (energy demanding) process of crushing rocks. In principle, however, any other model for the constraints and the objective fits the framework considered, as long as it is formulated in terms of linear equalities, or inequalities, involving the decision variable \( x \).

### 9.3.1 Reserve Provision Requirement – Introducing Structured Uncertainty in the Model

The solutions of the nominal model (9.12) are not adequate for providing reserves because there is no guarantee that shifting jobs' start times does not break some inequality constraints. This issue is exacerbated by the fact that jobs tend to be scheduled tightly at times of low prices. The geometrical interpretation is that solutions lie close to the boundary of the feasible set, and hence small perturbations can make them infeasible. The corresponding schedules are thus not flexible.
There are different approaches to solve this problem. One way is to recalculate a new schedule at the time when reserve is called, for instance enforcing lower consumption by lowering $\pi$. The problem is that in this case the new schedule may be completely different from the initial one, which is impractical.

Here we opt for an alternative way in which a flexible schedule is obtained using the robust optimization approach proposed in the previous Section 9.2. To see how the model (UIP) can be of use here (in particular the introduction of uncertainty that depends on the decision taken), we first consider a simplified example.

Suppose that the scheduling horizon has 4 steps, and only one job has to be scheduled. In the solution computed, the job is to be started at time step 3, i.e., $x^* = [0 \ 0 \ 1 \ 0]^T$. However, during execution, some external event forces a delay in the starting time by one time step, and the actual solution implemented is $\hat{x} = [0 \ 0 \ 0 \ 1]$. We can model this interaction as

$$\hat{x} = x + w = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix}.\$$

Clearly, the value of the disturbance $w$ depends on the choice $x^2$. Our goal here is to find a solution that will remain feasible for any such disturbance. We can cast this using the model (UIP), where the uncertainty structure (9.2) is to be written as

$$W(x) = \bigoplus_{k=1}^4 x[k] \cdot (\Omega_k \cdot \Lambda),$$

which can be parsed as follows. First, if $x[k] = 0$, then the uncertainty corresponding to the task being started at step $k$ is suppressed. On the other hand, when $x[k] = 1$, the uncertainty set from which $w$ is picked will be $\Omega_k \Lambda$.

$\Omega_k$ is a matrix containing all the different possible disturbance vectors as columns. Using our previous example, the matrix $\Omega_3$ could be, for instance,

$$\Omega_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & +1 & 0 \\ -1 & -1 & 0 \\ +1 & 0 & 0 \end{bmatrix},\quad (9.14)$$

in which the first column encodes a delay in the starting time by one time step, the second column an anticipation of execution by one step, and the third no event.

\(^2\)To give a further example of why this is the case, suppose we had $x^* = [0 \ 1 \ 0 \ 0]$ instead; then we would have had to encode a delay by one step as $w = [0 \ -1 \ +1 \ 0]$.\)
The set $\Lambda$ encodes the selection of a certain column from $\Omega_k$, and is the actual uncertain component of the problem (we do not know, when solving model (UIP), which column of $\Omega_k$ will be picked). Since we want to pick exactly one column, we define $\Lambda$ to be the feasible set of a multiple choice problem, i.e., for the example (9.14), we would need

$$\Lambda = \left\{ \lambda \in \mathbb{R}^3 \mid \sum_{j=1}^{3} w[j] = 1 \right\}. \quad (9.15)$$

Thus, what we would like to solve in this context is the following model

$$\begin{align*}
\begin{cases}
\min & c^T x \\
\text{subject to} & A(x + w) \leq b \quad w \in \mathcal{W}(x) \\
& x \in \{0,1\}^{N\times M},
\end{cases}
\end{align*}$$

which is a special case of (UIP) where $D = A$, the data $(A, b, c)$ is determined by the model (9.12) and $\mathcal{W}(x)$ is as in (9.13).

We now generalize to the case in which we have arbitrary many time steps in the horizon and tasks to be scheduled. The uncertainty set $\mathcal{W}(x)$ is then extended to

$$\mathcal{W}(x) = \oplus_{i,k} x[i][k] \cdot (\Omega_{(i,k)}^{R_i} \cdot \Lambda) \quad (9.16)$$

in which the summation index has been increased with the task index. We have also introduced $\Omega_{(i,k)}^{R_i} \in \mathbb{R}^{N \times R}$, which generalizes (9.14): it contains the vectors that shift the starting time of job $i$ in the future by at most $R_i$ time slots. We can automate its construction using

$$\Omega_{(i,k)}^{R_i}[i,j] = \begin{cases} 
-1 & \text{if } j = k \\
1 & \text{if } j = i - k \\
 & \text{and } k < i \leq (k + R_i) \\
0 & \text{otherwise},
\end{cases} \quad (9.17)$$

where $\Omega_{(i,k)}^{R_i}[i,j]$ is the $(i,j)$ entry of each matrix $\Omega_{(i,k)}^{R_i}$, and $R_i$ tunes the maximum amount of reserve to be provided in terms of time steps for each job $i$. We assume that the plant operator decides on the values for the $R_i$’s. An analogous construction is possible if negative reserve is to be provided, i.e. if margins to shift jobs at earlier times are wanted, in order to increase (rather than decrease) consumption temporarily.

The only step remaining to determine the flexible schedule is to compute the matrix $H$ according to Theorem 9.1. Given the above structure for $\mathcal{W}(x)$, the evaluation of the support function in (9.3) can be simplified as follows

$$H_{j,(i-1)N+k} = \sigma_{W_i} (a_j) = \sigma_{\Omega_{(i,k)}^{R_i} \cdot \Lambda} (a_j) = \sigma_{\Lambda} (a_j \cdot \Omega_{(i,k)}^{R_i}) = \sigma_{\text{conv}(\Lambda)}(a_j \cdot \Omega_{(i,k)}^{R_i})$$
i.e. to the maximization of a linear function over a simplex. The last step shows that the combinatorial structure in (9.15) is not an obstacle since we can consider its convex hull instead, of which we have direct access (conv(Λ) is just the linear relaxation of Λ). Note further that we do not need an external solver to compute this maximization, because its analytic solution is readily available: calculate the product $a_j \cdot \Omega^{R_i}_{(i,k)}$, and select the largest component of the resulting vector.

Note that the robustification procedure results in a change in the problem's data, as indicated by $(RC - UIP)$. This makes intuitive sense: for simple scheduling problems, which for instance do not have sequentiality nor earliest delivery, but just due time constraints, one could immediately construct a robust schedule by artificially increasing the process length. For instance, a task that would take one our to process, but is subject to a potential starting time delay of 1 hour, is just modelled as a task with a length of two hours.

What needs to be done on more complicated models is however not so straightforward. For instance, consider the case in which due time constraints are also to be included. We have modelled these as a limitation on the maximum amount of time that can elapse between the earliest possible starting time of task $i$, given its sequentiality conditions, and its chosen starting time (see Equation (9.10)). If the length of the task is just increased, the model does not know exactly when the task will start, which means it cannot ensure this maximum delay is guaranteed. In such a case it is useful to deploy the robustification procedure proposed, as it automates the computation of the new problem's data independently of how complicated the nominal model is.

### 9.4 Example: a Cement Producer

The electricity cost minimization model we use here is adapted from [Castro et al., 2011] and it concerns the scheduling the process of crushing rocks, which is a very energy-intensive process step of the cement production chain.

In the case presented in [Castro et al., 2011], there are two crushing machines whose scheduling for one week has to be established. Each one of the two machines consumes 5MW. While in [Castro et al., 2011] the decision of which quality of rocks to be crushed and in which quantity is also integrated in the problem, due to lack of data here this is assumed to be already determined and is taken into account with appropriate earliest delivery and due time constraints.

The scheduling model used is (9.12). Each of the two machines is scheduled separately, and there are no interdependencies among the jobs (both S matrices are identically equal
Robust Mixed-Integer Optimization, with Application to Uncertain Scheduling Problems

Table 9.1: Data used for scheduling Machine 1; values are in hours.

<table>
<thead>
<tr>
<th>Task #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l_i$</td>
<td>7</td>
<td>2</td>
<td>8</td>
<td>3</td>
<td>7</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>$d_i$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>$\bar{d}_i$</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>46</td>
</tr>
<tr>
<td>$R$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Task #</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>$l_i$</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>$d_i$</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>94</td>
<td>94</td>
</tr>
<tr>
<td>$\bar{d}_i$</td>
<td>72</td>
<td>72</td>
<td>72</td>
<td>72</td>
<td>96</td>
<td>96</td>
<td>96</td>
<td>96</td>
<td>96</td>
<td>120</td>
</tr>
<tr>
<td>$R$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Task #</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>$l_i$</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>13</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>$d_i$</td>
<td>94</td>
<td>94</td>
<td>118</td>
<td>118</td>
<td>118</td>
<td>118</td>
<td>118</td>
<td>142</td>
<td>142</td>
<td></td>
</tr>
<tr>
<td>$\bar{d}_i$</td>
<td>120</td>
<td>120</td>
<td>144</td>
<td>144</td>
<td>144</td>
<td>144</td>
<td>144</td>
<td>168</td>
<td>168</td>
<td></td>
</tr>
<tr>
<td>$R$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

to 0). Power consumption limits are $p = 0$ MW and $p = 5$ MW for both machines and for each time step. 17 jobs have to be scheduled on machine 1, 29 on machine 2, and each job consumes 5 MW (constant). Price profiles have been taken from the EEX website for the week going from the 27th of June to the 3rd of July 2011. Tables 9.1 and 9.2 contain the rest of the data used.

The resulting schedules obtained, both the nominal as well as the flexible ones, are depicted in Figure 9.2 and 9.3.

### 9.4.1 Bidding process

Reserves are sold on electricity markets for ancillary services using bids. The bidding process involves two decisions: how much reserve to offer and at what price.

The amounts are chosen by the operator when constructing the $\Omega_{(i,k)}^R$ matrices; more specifically when choosing the parameter $R$ for each job. As long as a feasible robust solution can be found, they can be bid.

The price demanded, on the other hand, has to be set high enough to at least cover the additional costs incurred. In contrast to load shedding, variable costs for load shifting do

---

3 See: http://www.eex.com/
Machine 2

<table>
<thead>
<tr>
<th>Task #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l_i$</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>$d_{ij}$</td>
<td>0</td>
<td>22</td>
<td>22</td>
<td>46</td>
<td>46</td>
<td>70</td>
<td>70</td>
<td>94</td>
<td>94</td>
<td>94</td>
</tr>
<tr>
<td>$d_{ij}$</td>
<td>24</td>
<td>48</td>
<td>48</td>
<td>72</td>
<td>72</td>
<td>96</td>
<td>96</td>
<td>120</td>
<td>120</td>
<td>120</td>
</tr>
<tr>
<td>$R$</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 9.2: Data used for scheduling Machine 2; values are in hours.

Figure 9.2: Power consumption profiles obtained by solving the nominal problem (9.12) and the explicit robust counterpart ($RC - UIP$), together with the price profile (taken from EEX).

not include “value of lost load” costs. These opportunity costs do not arise because the consumption is recuperated at a later time, and it is the main reason why variable costs are generally drastically lower than load shedding. For instance, in the case of paper production companies, variable costs of load shifting are estimated to be lower than 10 Euros/MWh versus costs of 1000 Euros/MWh and beyond for load shedding [Paulus and Borggreve, 2011].

The additional variable costs incurred when shifting load account for storage costs, electricity price costs, and additional contingent costs. Storage costs are typically negligible [Paulus and Borggreve, 2011], and contingent costs (such as the cost for having personal working shifted schedules) strongly depend on the particular production facility involved. Electricity price costs can be split into two: the costs incurred for resorting to the suboptimal flexible schedule and the premium for the risk involved into buying energy at spot prices whenever the reserve is called upon. The proposed algorithm gives
Figure 9.3: Nominal and robust schedules for the two milling machines of the case study. The dark shaded boxes are the jobs (arbitrarily) selected for reserve provision.

indications on the first part of these electricity costs: they can be obtained by comparing the nominal optimal objective value with the robust one. For the particular case of the cement milling machines presented earlier, the electricity costs are increased by approximately 1.7% (ca. 600 Euros). The estimation of electricity price risks, on the other hand, is an open question.
10. An Approach for Model Predictive Control of Mixed-Integer-Input Linear Systems based on Convex Relaxations

Model predictive control (MPC) [Maciejowski, 2002] is an optimization-based technique in which control inputs are applied in a receding horizon fashion. At every time step, the input applied is the first of a sequence computed by solving a finite horizon optimal control problem for the current state. The process is repeated at the next time step given a new measurement or estimate of the system state. MPC most commonly employs a linearized model of the system, and convex cost and constraint functions. In this case the corresponding finite horizon optimal control problem can be solved efficiently and reliably, since it is a convex optimization problem.

It is also possible to control systems with discrete state and/or input constraints by encoding a hybrid, rather than linear, system model into the MPC optimization problem [Bemporad and Morari, 1999]. However, this results in a non-convex problem with integer constraints, typically a mixed-integer quadratic program (MIQP). MIQPs must be solved using methods with exponential worst-case complexity, which limits the practical applicability.

In order to overcome this limitation, research so far has mainly focused on two approaches. In the first one, tailored methods to efficiently solve the corresponding mixed integer problems using branch and bound methods have been proposed. In particular, considerable research has been directed into efficiently computable and tight lower bounds to the optimal objective function, which are useful as they speed up the pruning of the B&B search tree [Axehill et al., 2007a, Axehill et al., 2007b, Axehill et al., 2010]. The other approach relies on the computation of explicit control laws [Francesco Borrelli, Alberto Bemporad, Manfred Morari, 2012]. These require only very limited computations in real time, but the memory required to store the explicit law increases dramatically with the problem size.

In this work we focus on the case in which integer constraints are imposed on the inputs only. In our proposed approach we solve a single convex relaxation of the original control problem, and use a projection function to project the resulting inputs onto the set of
inputs satisfying the integer constraints. However, the new feasible input sequence also causes changes in the state trajectory. In order to ensure that state constraints are also satisfied for the new input sequence, we use robust MPC techniques. We thus guarantee that the solutions provided are feasible both in terms of input and state constraints. Because our method only requires the solution to a convex optimal control problem followed by a computationally cheap projection, the time needed to compute a control input scales more modestly with the problem size than that needed to solve the associated MIQP exactly.

The key ingredient of the approach is the projection function. Such a function is often available in practice: we show how it can be designed for the important class of PWM systems, and we demonstrate the efficacy of the method by applying it to a constrained reference tracking problem for two different power converters. Our solutions offer significantly better performance than solutions based on the linearized model typically used for converter control.

### 10.1 Models Considered – Buck Converter Example

We use the buck converter (BC, Fig. 10.1) as an example to demonstrate the proposed idea. In the results Section we will apply our controller also to another power converter having more switches.

The main task of the BC is to convert a higher DC supply voltage $V_{\text{in}}$ by periodic switching to a lower DC output voltage $v_o(t)$. The system equations $\dot{x}(t) = A^c x(t) + B^c \delta(t)$ and the output equation $v_o(t) = C^c x(t)$ can be readily obtained from first principles, with

$$\dot{x}(t) = \begin{bmatrix} i_L(t) \\ v_C(t) \end{bmatrix}$$

(10.1)
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load</td>
<td>( R_{\text{out}} = 100 , \Omega )</td>
</tr>
<tr>
<td>Parasitic Resistances</td>
<td>( R_L, R_C = 0.1 , \Omega )</td>
</tr>
<tr>
<td>Capacitor</td>
<td>( C = 5 , \mu \text{F} )</td>
</tr>
<tr>
<td>Inductor</td>
<td>( L = 10 , \text{mH} )</td>
</tr>
<tr>
<td>Supply voltage</td>
<td>( V = 100 , \text{V} )</td>
</tr>
<tr>
<td>Switching period</td>
<td>( T_p = 0.1 , \text{ms} )</td>
</tr>
</tbody>
</table>

Table 10.1: Buck Converter: values of the parameters

and

\[
A^c = \begin{pmatrix}
-\frac{1}{L} & \frac{R_C R_{\text{out}}}{R_C + R_{\text{out}}} \\
\frac{1}{C} & -\frac{1}{C} \left( \frac{1}{R_C + R_{\text{out}}} \right)
\end{pmatrix}, \quad
B^c = \begin{pmatrix}
\frac{V_0}{L} \\
0
\end{pmatrix}, \quad
C^c = \begin{pmatrix}
\frac{1}{L} & \frac{1}{C}
\end{pmatrix}
\]

(10.2)

The switch position \( \delta(t) \in \{0, 1\} \) \( \forall t \) is the control input. The parameter values used in the simulations are given in Table 10.1.

### 10.1.1 Averaged Model

The standard way of controlling not only the BC but a wide range of switched systems in power electronics is by deriving an averaged model and apply Pulse Width Modulation (PWM) at the input.

In PWM, the time horizon is divided into switching periods and the binary input \( \delta(t) \) is replaced by \( d \in [0, 1] \) called the duty cycle. The duty cycle defines the time during which the switch is closed in the corresponding switching period and in PWM, an on-off signal is generated from it as depicted in figure 10.2 (A). The discrete-time dynamics can then be approximated linearly:

\[
\begin{align*}
x_{k+1} &= Ax_k + Bd_k, \quad 0 \leq d_k \leq 1 \\
y_k &= Cx_k
\end{align*}
\]

(10.3)

For the BC, the matrices \( A, B \) and \( C \) are obtained by discretizing (10.2). This simple model approximates the average values of the states over one switching period, but does not reflect the (hybrid) system dynamics within one cycle at all. This shortcoming is well known [Fischer et al., 2011] and results in several main drawbacks: constraint violations may occur, as depicted e.g. in Figure 10.3, the non-modeled hybrid dynamics of the system may lead to steady state deviations which is highlighted in Section 10.3 and no objective on the switching and in particular on the switching losses can be formulated.
Figure 10.2: The averaged model (A) assumes a constant control signal $v_k = d_k$ over one switching period (dashed curve), and this control is transformed into a 0/1 signal (solid curve) by a PWM. The hybrid model (B) generates a better approximation of the control signal $v_k$ (dashed curve), and this control must again be transformed into an appropriate on/off signal (solid curve).

An approach that has received attention lately [Fischer et al., 2011, Geyer et al., 2008, Papafotiou et al., 2004] is the use of more refined system models that capture the hybrid dynamics explicitly. These models divide the switching periods further into samples and approximate the system dynamics over these samples instead. Feasibility of the input, in particular allowing at most one on/off switch per switching period, is ensured using logical constraints. We describe this procedure in more detail in the next section.

### 10.1.2 Hybrid Model

The use of hybrid models for power converters is interesting since it makes it possible to pose control objectives dependent on the hybrid dynamics (e.g., current ripple, switching losses), which may lead to further energy efficiency improvements. Dealing with disturbances robustly may be important for safety or power quality reasons.

To obtain such model, we use an approach similar to the one described in [Fischer et al., 2011]. We divide each switching period of length $T_p$ into $M$ intervals of equal length $T_s = T_p/M$. To model the switching, we introduce new inputs $\delta_k^- \in \{0,1\}^M$ and $\delta_k^+ \in \{0,1\}^M$ for each switch. A value $\delta_{k,i}^+ = 1$ denotes that the switch is closed at some point during sample $i$ in switching period $k$, $\delta_{k,i}^- = 1$ means that is is opened. The exact
time when the switch is operated between sample $i$ and $i+1$ is modeled by a continuous input $d_k \in \mathbb{R}^M$ restricted to $-\delta_k^- \leq d_k \leq \delta_k^+$ in which "\leq" is interpreted element-wise. Thus the continuous signal can be different from 0 only during samples when a decision to switch on or off is made. For notational convenience, we introduce for each PWM operated switch the control signal $v_k \in \mathbb{R}^M$ defined as

$$v_k := \begin{pmatrix} 0 & \cdots & 0 \\ 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix} \cdot (\delta_k^+ - \delta_k^-) + d_k$$ (10.4)

which is illustrated in Figure 10.2 (B). The value $v_{k,i}$ indicates the fraction of sample $i$ in switching period $k$ during which the switch is closed.

We wish to control the state during the switching periods, and therefore introduce the stacked state vector

$$x_k^\top := (x_{k,0}^\top \ x_{k,1}^\top \ \cdots \ x_{k,M-1}^\top)$$ (10.5)

in which $x_{k,i} = x(k \cdot T_p + i \cdot T_s)$ denotes the states at the $i^{th}$ sample during the $k^{th}$ switching period. The system dynamics are given by

$$x_{k+1} = A \cdot x_k + B \cdot v_k$$ (10.6)

in which the matrices $A, B$ are given by

$$A = \begin{pmatrix} 0 & \cdots & 0 & A \\ & \ddots & \vdots & \ddots \\ & & 0 & \cdots & 0 \\ & & & A^M & \cdots \end{pmatrix}, \quad B = \begin{pmatrix} B & \cdots & 0 \\ & \ddots & \vdots \\ & 0 & \cdots & B \\ & 0 & \cdots & 0 \end{pmatrix}$$ (10.7)

with matrices $A, B$ obtained by discretizing the system dynamics (10.3) with sample time $T_s$.

In order to reproduce the switched behavior in our system model, constraints need to be imposed on the input. In the input constraints

$$0 \leq v_k \leq 1$$ (10.8a)

$$1^\top \cdot \delta_k^+ \leq 1$$ (10.8b)

$$1^\top \cdot \delta_k^- \leq 1$$ (10.8c)

$$-\delta_k^- \leq d_k \leq \delta_k^+$$ (10.8d)

we impose (10.8a) to ensure feasibility of the input, (10.8b) and (10.8c) ensure that we switch on resp. off at most once per cycle and (10.8d) models the exact switching time
as described before. Last, an integrality constraint needs to be imposed on the switching inputs

$$\delta^+_k, \delta^-_k \in \{0, 1\}^M$$

(10.9)

Together (10.8) and (10.9) are referred to as input constraints and we denote an input $$u_k$$ which satisfies the input constraints as input feasible.

In addition to the input constraints, we allow for general polytopic constraints on the states

$$E_x \cdot x_k \leq e.$$  

(10.10)

It should be highlighted that the maximal switching frequency of the hybrid model is the same as in the averaged model. Likewise, the control algorithm based on the hybrid model is evaluated only once per switching period. A system model which is updated at every sample is also possible using the more general Mixed Logical Dynamical (MLD) framework [Bemporad and Morari, 1999], but such an approach leads to a time varying system model. In addition, we explicitly know the set of input feasible inputs at each step for our model, hence it will be easy to find a suitable projection that provides inputs satisfying the integrality constraints. This is not possible for general MLD systems.

### 10.2 MPC formulation

In model predictive control, the control objective to be minimized is defined stage-wise as

$$\sum_{k=1}^{N} J_k(x_k, u_{k-1}),$$  

(10.11)

where each function $$J_k$$ is the cost at stage $$k$$, and $$N$$ is the number of stages in the problem. In the simulations of the BC we control the output voltage and thus define an objective for the averaged system model

$$J^{avg}_k(x_k, u_{k-1}) = (x_k^T - \bar{x}_k)^T C^T C (x_k^T - \bar{x}_k)$$  

(10.12)

in which $$C$$ denotes the system output matrix and $$\bar{x}_k$$ a reference. A similar objective $$J^{hyb}_k(x_k, u_{k-1})$$ can be defined for the hybrid system.

In MPC, an infinite horizon optimal control problem is approximated by a number of finite horizon optimal control problems, which are solved in receding horizon fashion. We can thus define MPC controllers for the averaged resp. hybrid system model by posing the
following finite horizon constrained optimal control problems for the averaged system:

\[
\begin{align*}
\min_{x_k, u_k} & \sum_{k=1}^{N} J_{\text{avg}}^k(x_k, u_{k-1}) \\
\text{s.t.} & \quad \text{averaged dynamics (10.3)} \\
& \quad \text{state constraints (10.10)}
\end{align*}
\]

(10.13)

and the hybrid system:

\[
\begin{align*}
\min_{x_k, u_k} & \sum_{k=1}^{N} J_{\text{hyb}}^k(x_k, u_{k-1}) \\
\text{s.t.} & \quad \text{hybrid dynamics (10.6)} \\
& \quad \text{state constraints (10.10)} \\
& \quad \text{switching constraints (10.8)} \\
& \quad \text{integrality constraints (10.9)}
\end{align*}
\]

(10.14)

The performance of the MPC controllers for the buck converter on a step response with a state constraint on the maximal current is depicted in Figure 10.3 for the MPC based on the averaged model and in Figure 10.4 for the hybrid system model. Violation of the state constraints is clearly visible under the action of the controller based on the averaged model. In addition, the model based on the duty cycle leads to steady state inaccuracies (see Section 10.4).

### 10.3 Relaxation based Hybrid MPC

In this section we state the relaxation of the hybrid optimal control problem (10.14). We propose a projection function for PWM inputs, and a method to establish – a priori – the influence of the projection operation on the states’ trajectories. This information can be used to robustify the relaxed optimization problem, which ensures that the projected sequence of inputs (which is input feasible by virtue of the projection function) does not violate state constraints either.

The most straightforward relaxation is the replacement of any integrality constraint \( x \in \{0, 1\} \) with the convex relaxation \( x \in [0, 1] \). The relaxation of the hybrid MPC optimal control problem is thus defined by (10.14) with a relaxed integrality constraint (10.9).

#### 10.3.1 Projection function

We propose a suitable projection function for system with PWM inputs next, in which \( \tilde{\delta}_k^+, \tilde{\delta}_k^- \) denote the relaxed counterparts of \( \delta_k^+, \delta_k^- \) and \( \tilde{d}_k \) is the counterpart to \( d_k \). All
relaxed inputs are collected in $\tilde{u}_k = (\delta_k^+, \delta_k^-, \tilde{d}_k)$. Following (10.4), we also introduce $\tilde{v}_k := D \cdot (\delta_k^+ - \delta_k^-) + \tilde{d}_k$.

We chose a projection such that the average duty cycle $\tilde{D} = \int_0^{T_s} v(t) dt = M^\top \cdot \tilde{v}_k$ and the first moment of the input $\tilde{M} = \int_0^{T_s} t \cdot v(t) dt = (1/2 \cdot 3/2 \ldots (M - 1/2)) \cdot \tilde{v}_k$ over one switching period are preserved. The only control actions we are allowed to make during one cycle is when to close the switch and when to open it, so intuitively speaking we have only two degrees of freedom. The switching times

$$t_{\text{on/off}} = \frac{\tilde{M}}{\tilde{D}} \pm \frac{\tilde{D}}{2}$$

(10.15)

can thus be computed explicitly.

The input defined by the switching times $t_{\text{on/off}}$ can be encoded in a hybrid input $u_k$ and we denote by $p(\tilde{u}_k) : \mathbb{R}^M \times \mathbb{R}^M \times \mathbb{R}^M \to \{0, 1\}^M \times \{0, 1\}^M \times \mathbb{R}^M$ the projection function for PWM inputs that projects the relaxed input $\tilde{u}_k$ to a hybrid input $u_k$ such that the switching times are defined as above. This hybrid input is input feasible by construction.
Projected inputs $\mathbf{u}_k = p(\bar{\mathbf{u}}_k)$ satisfy the following constraints:

\begin{align}
\mathbf{1}_M^\top \cdot \mathbf{v}_k &= \mathbf{1}_M^\top \cdot \bar{\mathbf{v}}_k \quad (10.16a) \\
\begin{bmatrix} 1/2 & 3/2 & \cdots & (M - 1/2) \end{bmatrix} \cdot (\mathbf{v}_k - \bar{\mathbf{v}}_k) &= 0 \quad (10.16b) \\
\text{input feasibility (10.8), (10.9)} \quad (10.16c)
\end{align}

Examples of the projection of several relaxed inputs $\bar{\mathbf{u}}_k$ are given in Figure 10.5. Note that by using this projection, the switch will be open at the end of every cycle.

### 10.3.2 Robust feasibility

We want to implement a controller which solves the QP-relaxation of the MIQP, uses the previously defined projection and applies the projected inputs to the plant.

Applying the projected inputs $\mathbf{u}_k = p(\bar{\mathbf{u}}_k)$ instead of the relaxed inputs $\bar{\mathbf{u}}_k$ to the system will yield a different state evolution in general. We refer to the set of all possible state deviations as the induced uncertainty set $\mathcal{W}$, i.e.

\begin{align}
\mathcal{W} &:= \{ \Delta \mathbf{x} \mid \Delta \mathbf{x} \text{ satisfies (\star) for some } \bar{\mathbf{u}}_k, \mathbf{x}_k \} \\
(\star) \quad \Delta \mathbf{x} &= \mathbf{x}_{k+1}(\rho(\bar{\mathbf{u}}_k), \mathbf{x}_k) - \mathbf{x}_{k+1}(\bar{\mathbf{u}}_k, \mathbf{x}_k) \quad (10.17)
\end{align}
10 An Approach for Model Predictive Control of Mixed-Integer-Input Linear Systems based on Convex Relaxations

Figure 10.5: The projection function for the BC, with $M = 5$ samples of length $T_s$ per switching period $T_p$. Depicted are the binary input and the control signal $\bar{v}_k$. For each switching period, the area below the graphs is the same (10.16a) and that the first moments of the inputs coincide (10.16b).

For any relaxed input $\bar{u}_k$ the state deviation can be written as

$$x_{k+1}(\bar{u}_k, x_k) - x_{k+1}(\bar{u}_k, x_k) = A \cdot x_k + B \cdot \rho(\bar{u}_k) - (A \cdot x_k + B \cdot \bar{u}_k) \quad (10.18)$$

i.e. the uncertainty set is independent of the state $x_k$ since we assume linear system dynamics. We can interpret the state deviation induced by the projection as unknown disturbances which are bounded within the uncertainty set, i.e.

$$x_{k+1} = Ax_k + B \rho(\bar{u}_k) = Ax_k + B\bar{u}_k + w_k \quad (10.19)$$

for some $w_k \in W$. But this equivalence makes use of robust MPC approaches for additive polytopic uncertainty possible in order to achieve robust feasibility of the computed inputs. We use a constraint tightening approach [Richards and How, 2006] in order to achieve robust feasibility. In this approach, the nominal feasible set $\{x \mid E_x x \leq e\}$, i.e. all states that satisfy (10.10), is tightened for subsequent iterations in the prediction horizon to account for the disturbances. For open loop robust control constraints defining those sets are given by

$$E_x x_k \leq e - \max_{w \in W} E_x \left( \sum_{i=0}^{k-1} A^i w_i \right) := e_k \quad (10.20)$$

which means that only a modification of the RHS of the state constraints is necessary, at no additional online computational effort. This modification makes the precomputed inputs robustly feasible meaning that all state and input constraints are guaranteed to be satisfied up to the prediction horizon for the projected inputs. We do not need to
compute the uncertainty set $\mathcal{W}$ explicitly, but we can instead evaluate the RHS of the modified state constraints (10.20) offline. To this end, we define

$$B := \begin{pmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1} B & \cdots & AB & B \end{pmatrix}, \quad \mathcal{E} := \mathbb{I}_N \otimes \mathbf{e}_x \quad (10.21)$$

where "$\otimes$" denotes the Kronecker product, and rewrite the RHS of the modified constraints as

$$\begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{pmatrix} = \begin{pmatrix} e \\ e \\ \vdots \\ e \end{pmatrix} - \max_{\bar{u}_k \text{ feasible}} \mathcal{E} \cdot B \cdot \begin{pmatrix} p(\bar{u}_1) \\ p(\bar{u}_2) \\ \vdots \\ p(\bar{u}_N) \end{pmatrix} \quad (10.22)$$

This optimization problem can be written in tractable form as a mixed integer linear program (MILP)

$$\max_{\bar{u}, \bar{\bar{u}}} \mathcal{E} \cdot B \cdot \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \end{pmatrix} - \begin{pmatrix} \bar{u}_0 \\ \bar{u}_1 \\ \vdots \\ \bar{u}_N \end{pmatrix} \quad (10.23a)$$

s.t. $1_M^{\top} \cdot \nu_k = 1_M^{\top} \cdot \bar{\nu}_k \quad \forall k \quad (10.23b)$

$$(1/2 \ 3/2 \ \ldots \ (M-1/2)) \cdot \nu_k \ldots = (1/2 \ 3/2 \ \ldots \ (M-1/2)) \cdot \bar{\nu}_k \quad (10.23c)$$

$$1_M^{\top} \cdot (\delta^+_k - \delta^-_k) = 0 \quad (10.23d)$$

$$1_M^{\top} \cdot (\bar{\delta}^+_k - \bar{\delta}^-_k) = 0 \quad (10.23e)$$

$u_k$ satisfy (10.8), (10.9) \quad (10.23f)

$\bar{u}_k$ satisfy (10.8) \quad (10.23g)

with "min" meaning row wise minimization. Each row of (10.23a) gives the amount by which the RHS value of one constraint for one stage in the prediction horizon needs to be tightened.

- Conditions (10.23b) and (10.23c) define the projection
- Conditions (10.23d) and (10.23e) ensure that the switch is open at the end of every cycle
- Conditions (10.23f) and (10.23g) ensure input feasibility

The maximal state deviations for the BC over two switching cycles are depicted in Figure 10.6. All computations for the constraint tightening can be done offline and need to be done only once. Thus it does not matter that several computationally expensive MILPs need to be solved.
Figure 10.6: The uncertainty set $W$ for the states of the BC, $M = 5$. The uncertainty is given for each sample in a horizon of two cycles. Note that the uncertainty becomes small at the end of every cycle.

In summary, the proposed MPC based on the projection approach uses the following finite horizon optimal control problem

$$
\min_{x_k, \tilde{u}_k} \sum_{k=1}^{N} j_{k}^{map}(x_k, \tilde{u}_{k-1}) = \sum_{k=1}^{N} j_{k}^{hyb}(x_k, \tilde{u}_{k-1})
$$

s.t. modified dynamics (10.6)
modified state constraints (10.20)
switching constraints (10.8)
relaxed integrality constraints $0 \leq \delta_k^+, \delta_k^- \leq 1$  \hfill (10.24)

10.4 Evaluation

We apply the MPC controller based on the averaged model (10.24) to control the Buck Converter (Figure 10.7). The resulting step response is illustrated in Figure 10.3. Constraint violations, which were present when using a traditional MPC based on the averaged model, are solved. In addition, the classical PWM model may lead to inaccuracies in the steady state behavior [Fischer et al., 2011]. The MPC controllers based on the hybrid model, both the hybrid and the relaxed one, are able achieve a better steady state RMS
Figure 10.7: Step response under MPC for the buck converter based on a QP-relaxation of the hybrid system model. Depicted are the states and the inputs, i.e. the precomputed input by the controller (green dashed) and the binary input that is actually applied to the system. The constraint on $i_L$ is satisfied and the steady state error is very small.

deviation from the reference. The corresponding state trajectories are depicted in Figure 10.9. The steady state RMS deviation of the continuous time output voltage from the reference and the online computation times are summarized in Table 10.2. The optimization problems have been solved using Yalmip [Löfberg, 2004] and Gurobi [Gurobi, 2014].

In order to demonstrate that our approach can be readily used with more complex PWM topologies, the proposed MPC approach has also been applied to a single phase grid inverter, which is used to connect a DC source or load to an AC grid. The single phase grid inverter is a linear system operated by four switches (Figure 10.8), out of which two ($s_1$ and $s_2$) can be operated independently by PWM. Using the parameters and the linear model described in [Fischer et al., 2013], MPC controllers based on the averaged model, the hybrid model and the projection approach have been implemented. We operate the grid inverter such that power from the DC source is injected into the grid, which is achieved by tracking a sinusoidal reference current of 50 Hz with an amplitude of 20 A. The resulting RMS deviation of the continuous time output (grid current) from the reference and computation times are reported in Table 10.2. In addition to achieving a better RMS deviation than MPC based on the averaged model, relaxation based MPC
Figure 10.8: Single phase grid inverter circuit.

<table>
<thead>
<tr>
<th><strong>Buck Converter</strong></th>
<th>average 1.6 V</th>
<th>hybrid 0.21 V</th>
<th>QP-relaxation 0.27 V</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMS deviation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>comp. time, N = 2, M = 5</td>
<td>1.6 ms</td>
<td>9.2 ms</td>
<td>3.4 ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>AC-DC converter</strong></th>
<th>average 0.415 V</th>
<th>hybrid 0.098 V</th>
<th>QP-relaxation 0.185 V</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMS deviation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>comp. time, N = 4, M = 3</td>
<td>2.6 ms</td>
<td>53.3 ms</td>
<td>6.1 ms</td>
</tr>
</tbody>
</table>

Table 10.2: MPC performance

achieves superior computation times in comparison to hybrid MPC.
Figure 10.9: Sampling times are indicated by 'o', the voltage $u_{out}$ is the system output.
(a) Using the averaged model, model inaccuracies and the current ripple which is unknown to the controller lead to a significant steady state offset.
(b) The projection based MPC achieves a better approximation of the system behavior through a more refined system model and reduces the RMS deviations significantly even in the presence of the uncertainty induced by the projection.
11. Extending Affine Control Policies to the Robust Control of Hybrid Systems

Robust Model Predictive Control (MPC) is concerned with controlling a system subject to external disturbances and/or parameter uncertainties, while ensuring that hard constraints on states and inputs are satisfied for any possible disturbance from some bounded set. Control decisions are planned on a receding horizon basis, in which at every time step a sequence of actions is chosen. It is well established that for Robust MPC the possibility of recourse on the decisions planned within the optimization horizon has to be included in order to avoid excessive (and unnecessary) conservatism or, even worse, feasibility problems [Goulart et al., 2006]. Parameterizing such recourse using causal, affine functions of disturbances measured during the time horizon has emerged as a good compromise between performance and tractability [Löfberg, 2003, Goulart et al., 2006]. Those results were facilitated by advances in solving robust optimization problems where the minimizer was allowed to be a function of the uncertain problem data [Ben-Tal et al., 2004b].

Such affine recourse functions have so far only been studied in detail for the control of linear systems. In this paper we extend the idea of using affine policies to hybrid systems, by proposing a robust controller that includes recourse.

In order to demonstrate the efficacy of our proposed controller on a switched dynamical system, we test it on the DC-DC buck power converter (also known as the step-down DC-DC converter), a device that was introduced in Section 10.1 of the previous chapter. For this application, both linear as well as hybrid models exist, and both have been used in a predictive control context [Papafotiou et al., 2004, Geyer et al., 2008, Fischer et al., 2011]. We have provided both of these models in Section 10.1. Using the linear model, in Section 11.2 we apply a standard robust controller based on the affine policy approach of [Goulart et al., 2006] and highlight its limitation, namely that the linear model is a rather coarse approximation and does not capture the switched behaviour of the system. This leads to violations of hard constraints despite the use of a robust controller. This problem motivates the use of hybrid models.

We demonstrate a new controller that chooses integer decision variables taking into account that recourse will be taken on the continuous variables, thereby eliminating
problems arising from the use of a standard averaged model. Results from the two controllers are compared, and significant performance improvement is reported for the new controller.

11.1 Robust hybrid optimal control problem

In this section we pose a general robust hybrid optimal control problem. We consider Mixed Logical Dynamical (MLD) systems, which were introduced in [Bemporad and Morari, 1999]. MLD systems encompass a wide variety of systems, including switched systems such as the buck converter. The discrete-time dynamics at a time step \( k \) are described by

\[
    x_{k+1} = Ax_k + Bu_k + B_2\delta_k + B_3z_k + Gw_k. \tag{11.1}
\]

The variables \( x_k \in \mathbb{R}^n, u_k \in \mathbb{R}^{n_u} \) and \( w_k \in \mathbb{R}^{n_w} \) are respectively the states, inputs, and disturbances. The variables \( z_k \in \mathbb{R}^{n_z} \) and \( \delta_k \in \{0, 1\}^{n_\delta} \) are auxiliary variables characterizing the hybrid behaviour of the system. They enter both in the dynamics, where they can express for instance switching between multiple modes, as well as in the constraints, described below, where they can be used in order to express logical conditions. In general, both continuous \((z_k)\) as well as discrete variables \((\delta_k)\) are necessary to express such conditions.

We restrict our attention to systems which at each time step \( k \) are perturbed by a disturbance \( w_k \) bounded in a polytopic set \( \mathcal{W}_k \), which is described by the matrix \( S_k \in \mathbb{R}^{n_w \times n_w} \) and the vector \( h_k \in \mathbb{R}^{n_h} \) as follows:

\[
    \mathcal{W}_k = \{ w_k \in \mathbb{R}^{n_w} \mid S_kw_k \leq h_k \}. \tag{11.2}
\]

Hard state and input constraints are to be satisfied for any possible disturbance in \( \mathcal{W}_k \) at every time timestep \( k \). They are defined as follows:

\[
    \begin{align*}
        E_x x_k + E_u u_k + E_z z_k + E_\delta \delta_k &\leq e \\
        E_f x_N &\leq e_f
    \end{align*}
\]

\( \forall w_k \in \mathcal{W}_k \) \tag{11.3}

where we have introduced vectors \( e \in \mathbb{R}^{n_x} \) and \( e_f \in \mathbb{R}^{n_f} \), and the matrices \( E_x, E_u, E_z, E_\delta \) and \( E_f \) of appropriate dimensions. Note that since \( \delta \in \{0, 1\}^{n_\delta} \) is integer, the feasible region for the optimal control problem is in general non-convex.

We now express the Equations (11.1) and (11.3) in stacked form in order to formulate a finite-horizon optimal control problem. We first introduce the variables

\[
    \begin{align*}
        x &\doteq \begin{bmatrix} x_0^T, \ldots, x_N^T \end{bmatrix}^T \\
        u &\doteq \begin{bmatrix} u_0^T, \ldots, u_{N-1}^T \end{bmatrix}^T \\
        z &\doteq \begin{bmatrix} z_0^T, \ldots, z_{N-1}^T \end{bmatrix}^T \\
        \delta &\doteq \begin{bmatrix} \delta_0^T, \ldots, \delta_{N-1}^T \end{bmatrix}^T \\
        w &\doteq \begin{bmatrix} w_0^T, \ldots, w_{N-1}^T \end{bmatrix}^T
    \end{align*}
\]

\( \tag{11.4} \)
11 Extending Affine Control Policies to the Robust Control of Hybrid Systems

the matrices

\[
A = \begin{pmatrix}
I \\
A \\
A^2 \\
\vdots \\
A^N
\end{pmatrix},
\quad
B = \begin{pmatrix}
0 & \cdots & 0 \\
B & 0 \\
AB & B & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots \\
A^{N-1}B & \cdots & AB & B
\end{pmatrix},
\tag{11.5}
\]

and the matrices \(B_2, B_3, \) and \(G,\) constructed exactly as for \(B\) but with \(B\) replaced by \(B_2,\) \(B_3,\) and \(G\) respectively. The state-update Equations (11.1) in stacked form then become

\[
x = Ax_0 + Bu + B_2\delta + B_3z + Gw.
\tag{11.6}
\]

For the constraints, we introduce the matrices \(E_x, E_u,\) and the vector \(e\) defined by

\[
E_x = \begin{pmatrix} I_N \otimes E_x & 0 \\ 0 & E_f \end{pmatrix},
\quad
E_u = \begin{pmatrix} I_N \otimes E_u & 0 \\ 0 & e_f \end{pmatrix},
\quad
e = \begin{pmatrix} 1_N \otimes e \\ e_f \end{pmatrix},
\tag{11.7}
\]

where the operation \(\otimes\) indicates the Kronecker product, and \(E_f\) and \(e_f\) are used to define terminal state constraints, i.e. constraints on \(x_N.\) The matrices \(E_3\) and \(E_2\) are defined analogously to \(E_u.\)

We also define the stacked uncertainty set \(W = \{w \in \mathbb{R}^{Nn} | Sw \leq h\},\) where \(S = \text{blkdiag}(S_0, S_1, \ldots, S_{N-1})\) and \(h = (h_0^T, h_1^T, \ldots, h_{N-1}^T)^T.\)

Constraint (11.3) can then be expressed in stacked form as

\[
E_x x + E_u u + E_3 \delta + E_2 z \leq e \quad \forall w \in W.
\tag{11.8}
\]

In this paper, we consider the objective of minimizing the expected value of a quadratic state and input cost function over the horizon. This cost is defined relative to some desired set point \(x_{\text{ref}}.\) Let \(P \in \mathbb{R}^{nx \times nx},\) \(P_N \in \mathbb{R}^{nx \times nx}\) and \(Q \in \mathbb{R}^{nu \times nu}\) be positive semidefinite symmetric matrices defining the state, terminal state and input stage costs respectively. Let \(P = \text{blkdiag}(P, \ldots, P, P_N)\) and \(Q = \text{blkdiag}(Q, \ldots, Q)\) be stacked versions of these costs matrices. We can then write the resulting finite horizon robust optimal control problem:

\[
\begin{aligned}
\min_{x, u, \delta, z} & \quad \mathbb{E} \left[(x - x_{\text{ref}})^T P (x - x_{\text{ref}}) + u^T Q u\right] \\
\text{s.t.} & \quad x = Ax_0 + Bu + B_2\delta + B_3z + Gw \\
& \quad E_x x + E_u u + E_3 \delta + E_2 z \leq e, \quad \forall w \in W \\
& \quad \delta \in \{0, 1\}^{N_n}.
\end{aligned}
\tag{ROCP}
\]

We will now investigate different ways of choosing inputs \(u\) in order to achieve good performance and constraint satisfaction for the specific example of a buck converter.
11.2 Affine policies based on the averaged model

In this section we highlight the limitations of the controller based on affine policies as it is applied to the linear model of the buck converter (i.e., the averaged model, see Section 10.1.1). The results obtained are unsatisfactory in that currents and voltages within the circuit do not satisfy the hard constraints.

We start by introducing the idea of affine recourse by parameterizing the sequence of controls \( u \) by the disturbances observable up to the time when each input is applied:

\[
    u_i = \sum_{j=0}^{i-1} M_{i,j} w_j + v_i \quad i = 1, \ldots, N - 1.
\]  

This can be rewritten in stacked form as

\[
    u = \begin{pmatrix} 0 & \cdots & 0 \\ M_{1,0} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ M_{N-1,0} & \cdots & M_{N-1,N-2} \end{pmatrix} \begin{pmatrix} w_0 \\ \vdots \\ w_{N-1} \end{pmatrix} + \begin{pmatrix} v_0 \\ \vdots \\ v_{N-1} \end{pmatrix}
\]

The robust optimal control problem (ROCP) specialized to a linear system, and with the substitution above becomes

\[
    \min_{M,v} \mathbb{E} \left[ (x - x_{\text{ref}})^T P (x - x_{\text{ref}}) + u^T Qu \right] \\
    \text{s.t.} \quad x = A x_0 + B(Mw + v) + Gw \\
    \quad E_x x + E_{x_1} (Mw + v) \leq e \quad \forall w \in \mathcal{W}
\]  

In Example 7 of [Goulart et al., 2006], Goulart et al. borrow an argument based on duality to show that a solution to this semi-infinite problem (the requirement that the constraints are satisfied \( \forall w \in \mathcal{W} \) implies an infinite number of constraints) can be obtained by solving the following finite and convex optimization problem

\[
    \min_{M,v} \mathbb{E} \left[ (x - x_{\text{ref}})^T P (x - x_{\text{ref}}) + u^T Qu \right] \\
    \text{s.t.} \quad E_x x_0 + E_{x_1} B w + E_{x_2} v - e \leq -\Lambda^T h \\
    \quad \Lambda^T S = (E_x G + E_{x_1} BM + E_{x_2} M) \\
    \quad \Lambda \geq 0
\]

in which all inequalities are component-wise and in which the new optimization variable \( \Lambda \in \mathbb{R}^{(N-n_h)\times(N-n_e+n_f)} \) has been introduced.

We use solutions to \( (\text{MPC}_{\text{avg}}) \) in a receding horizon fashion in order to control the buck converter. These optimization problems are constructed using Yalmip [Löfberg, 2004].
and solved using CPLEX 12.1 [IBM, 2014]. A realistic response of the converter is obtained using PLECS [Plexim, 2014], and Fig. 11.1 reports the results of a simple step response (scenario 1). The disturbances in the output current that the controller has to counteract are relatively large - up to 50% of the nominal output current. For the simulations, we use uniform, uncorrelated disturbances. In Fig. 11.1 we can directly see the problems involved with the linear model of the plant. Despite it being controlled by a robust controller, the constraint $i_L \leq i_{L_{\text{max}}} = 2i_{\text{nom}}$ is met in general only at the sampling times, marked “o” in the plot, while between the sampling times the current exceeds the bound. This motivates our investigation of robust controllers for hybrid models.

### 11.3 Mixing open-loop decisions with recourse

We now return to the hybrid model of the buck converter and develop a robust controller that still incorporates recourse. Affine policies are an appropriate choice in a context in which state and input constraints are convex, because they can map disturbances $w$ (which are drawn from a convex polytopic set) to control inputs $u$ (which are also constrained in a convex polytopic set). But if integrality conditions are imposed on some of the variables (in our case, $\delta_k^+, \delta_k^- \in \{0, 1\}$), an affine mapping from disturbances to the controls will in general produce controls that do not satisfy these integrality constraints.

In order to overcome this, we divide the generic control inputs $u$ appearing in (11.8) into
binary inputs \( d_k \in \{0,1\}^{n_d} \) and continuous inputs \( u_k \in \mathbb{R}^{n_u} \). Recourse is then allowed only for the continuous part \( u \) of the controls. We thus rewrite the state update equation such that the inputs are split into a continuous part \( u_k \) and binary part \( d_k \)

\[
x_{k+1} = Ax_k + B_u u_k + B_d d_k + G w_k,
\]

in which \( A, x, G \) and \( w_k \) are as before and the new input and input matrices are

\[
B_u := \begin{pmatrix} B_1 & 0 \\ B_2 & 0 \\ 0 & 1 \end{pmatrix}, \quad B_d := \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}
\]

\[
u_k := u_c(k), \quad d_k := \begin{pmatrix} \delta_+ \\ \delta_- \end{pmatrix}
\]

The stacked matrices \( B_u \) and \( B_d \) are defined analogously to \( \mathbf{B} \) in (11.5). Using this notation, the following optimization problem, referred to as the MPC_{ol/kl} problem, can be defined:

\[
\min_{v, M, d, \delta, z} \mathbb{E} \left[ (x - x_{ref})^T P (x - x_{ref}) + \left( \begin{pmatrix} u^T \\ d^T \end{pmatrix} Q \begin{pmatrix} u \\ d \end{pmatrix} \right) \right]
\]

s.t. \( x = A x_0 + B_u u + B_d d + B_2 \delta + B_3 z + G w \)

\( E_x x + E_u u + E_d d + E_\delta \delta + E_z z \leq e, \quad \forall w \in \mathcal{W} \)

\( u = M w + v \)

\[
\delta \in \{0,1\}^{N-n_d}, \quad d \in \{0,1\}^{N-n_d}.
\]

As we wish to parameterize the continuous inputs \( u \) by affine functions, we chose \( u = M \cdot w + v \). Feasible values for the binary variables \( d, \delta \) have to be found without recourse, thus we leave them as optimization variables in the equations. Note that since we only use recourse on the continuous variables, this optimization problem does not necessarily have a solution even though a feasible solution to the finite horizon problem in terms of a general non-linear control policy as found e.g. by dynamic programming, may exist. To simplify notation, we introduce

\[
\hat{x} = A x_0 + B_u v + B_d d + B_2 \delta + B_3 z \\
\hat{e} = E_x \hat{x} + E_u v + E_d d + E_\delta \delta + E_z z - e \\
\hat{f} = (\hat{x} - x_{ref})^T P (\hat{x} - x_{ref}) + \left( \begin{pmatrix} v^T \\ d^T \end{pmatrix} Q \begin{pmatrix} v \\ d \end{pmatrix} \right)
\]

Vector \( \hat{x} \) is the predicted state in case of zero disturbances, while \( \hat{e} \) equals the RHS of the system constraints and \( \hat{f} \) denotes the objective in this case. Note that they all depend on the nominal inputs \( v \), which are chosen open loop, but not on the disturbance dependent
recourse $M \cdot w$. We can thus rewrite (11.13) as

$$\begin{align*}
\min_{v, M, d, \delta, z} & \quad \mathbb{E} \left[ (x - x_{\text{ref}})^T P (x - x_{\text{ref}}) + (u^T d^T) Q \begin{pmatrix} u \\ d \end{pmatrix} \right] \\
\text{s.t.} & \quad \dot{e} + (E_{\delta} B_{\delta} M + E_{\delta} G + E_{\delta} M) w \leq 0, \quad \forall w \in \mathcal{W} \\
& \quad \dot{x}, \dot{e} \text{ as defined above} \\
& \quad \delta \in \{0, 1\}^{N_{\delta}}, \quad d \in \{0, 1\}^{N_{n_u}}.
\end{align*}$$

(11.15)

As in the case of the controller derived in Section 11.2, this problem must be rewritten in tractable form by replacing the condition $\forall w \in \mathcal{W}$ with a finite one. This is achieved starting with the following equivalences:

$$\begin{align*}
\dot{e} + (E_{\delta} B_{\delta} M + E_{\delta} G + E_{\delta} M) w & \leq 0, \quad \forall w \in \mathcal{W} \\
\Leftrightarrow \dot{e} & \leq -\max_{w \in \mathcal{W}} (E_{\delta} B_{\delta} M + E_{\delta} G + E_{\delta} M) w
\end{align*}$$

(11.16)

in which the maximization is row-wise. Following the idea in [Goulart et al., 2006], strong duality in linear programming gives the following equivalence:

$$\begin{align*}
\max_{w \in \mathcal{W}} (E_{\delta} B_{\delta} M + E_{\delta} G + E_{\delta} M) w & \quad \text{s.t.} \quad Sw \leq h \\
= \min_{\Lambda \geq 0} & \quad \Lambda^T h \quad \text{s.t.} \quad \Lambda^T S = E_{\delta} B_{\delta} M + E_{\delta} G + E_{\delta} M
\end{align*}$$

(11.17)

where $\Lambda$ is a matrix of new optimization variables whose entries are the Lagrange multipliers associated with the row-wise maximizations; (11.16) is thus equivalent to

$$\begin{align*}
\dot{e} & \leq -\Lambda^T h, \\
\Lambda^T S & = E_{\delta} B_{\delta} M + E_{\delta} G + E_{\delta} M, \\
\Lambda & \geq 0 \quad \text{element-wise.}
\end{align*}$$

(11.18)

Having rewritten the constraint vector in finite form, we note that the objective function can be rewritten as follows, assuming zero mean disturbances $\mathbb{E}[w_k] = 0$:

$$\begin{align*}
\mathbb{E} \left[ (x - x_{\text{ref}})^T P (x - x_{\text{ref}}) + (u^T d^T) Q \begin{pmatrix} u \\ d \end{pmatrix} \right] \\
= (\dot{x} - x_{\text{ref}})^T P (\dot{x} - x_{\text{ref}}) + (v^T d^T) Q \begin{pmatrix} v \\ d \end{pmatrix} \\
\ldots + \mathbb{E}[w^T D w]
\end{align*}$$

(11.19)

where $C_w \doteq \mathbb{E}[ww^T]$, $\dot{x}$ as before and $D \doteq (B_{u} M + G)^T P (B_{u} M + G) + M^T Q_{u} M$ and $Q_v \doteq Q(1:N_{n_u}, 1:N_{n_u})$. Collecting constraints and objective, we can now pose the MPC problem.
optimization problem for the buck converter in tractable form:

\[
\begin{align*}
\min_{v,M,d,\delta} & \quad \hat{f} + \text{trace}(D \cdot C_w) \\
\text{s.t.} & \quad \hat{e} \leq -\Lambda^T h, \\
& \quad \Lambda^T S = E_x B_x M + E_x G + E_u M \\
& \quad \Lambda \geq 0 \quad \text{element-wise} \\
& \quad \delta \in \{0,1\}^{N_n}, d \in \{0,1\}^{N_d} \\
& \quad \hat{f}, \hat{x}, \hat{e}, D \text{ as defined above.}
\end{align*}
\]

(MPC_{ol/cl})

### 11.4 Evaluation

In this section we compare three controllers; MPC\_avg based on the average model, MPC\_ol/cl based on the hybrid model, and MPC\_ol where recourse is excluded (i.e. \(M = 0\)) from the hybrid controller, and only an open-loop sequence is computed at each time step. In order to make this comparison, we let the two controllers run under two different experiments.

The first experiment is a step response, in which the reference voltage \(V_{a,ref}\) is raised to the desired reference of 33V. The box constraints are the same as those used to test MPC\_avg in Section 11.2, i.e. box constraints on the inductor current \(0 \leq i_L(k) \leq 2 \cdot i_{L,ref}\) but no constraint on the capacitor voltage, such that the comparison is fair. The results of this experiment are depicted in Fig. 11.2. We verify that MPC\_ol/cl ensures hard constraints satisfaction on the current. The picture also shows the recourse margins that can be used by the controller to modify its decision on when to operate the switch, depending on the disturbances measured. Note that the controller requires measurements taken at a higher frequency than the robust controller based on the averaged model, but the frequency at which the switch is operated is the same. This is crucial to limit switching losses.

In the second test, the task is to track a periodic trajectory with some added constraints to test the transient performance. The results are shown in Fig. 11.3, in which the response of both MPC\_avg as well as MPC\_ol/cl are depicted. The open-loop hybrid MPC MPC\_ol, on the other hand, has been tried but does not appear in the figure: the excessive conservativeness of this controller leads to infeasibility in the presence of reference changes and constraints. Note that this experiment is not designed to be unreasonably hard since even the simple linearized MPC\_avg is able to handle it. Table 11.1 summarizes the performance results.
11 Extending Affine Control Policies to the Robust Control of Hybrid Systems

Figure 11.2: Step response using MPC\textsubscript{ci/ci}, based on the hybrid model of the converter. Hard constraints on the current $i_L(k) \leq 2i_{L,\text{ref}}$ are satisfied, and the controller also has better tracking performance (see Table 11.1).

Figure 11.3: Comparison of reference tracking performance of the two controllers studied in this paper. The new proposed controller has access to a more refined model of the system, and thus achieves better tracking performance. Even for such a simple experiment, without recourse the optimization problem involved in the computation of the robust controller is infeasible.
Table 11.1: Performance of the controllers.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Steady state: RMS deviation of $V_o(t)$</th>
<th>Reference tracking: RMS deviation of $V_o(t)$, 2nd test</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC$_{avg}$</td>
<td>2.09V</td>
<td>15.2[V]</td>
</tr>
<tr>
<td>MPC$_{ol}$</td>
<td>0.89V</td>
<td>Infeasibility encountered</td>
</tr>
<tr>
<td>MPC$_{ol/cl}$</td>
<td>0.88V</td>
<td>9.74[V]</td>
</tr>
</tbody>
</table>
A. Examples and Counterexamples

A.1 Primal Recovery from $\lambda^*$ in the Non-Convex Case

In this section we discuss an example in which infeasibility issues related to primal recovery are particularly acute: every solution recovered is infeasible.

Consider the following problem:

\[
\begin{align*}
\text{minimize} & \quad -x_1 \\
\text{subject to} & \quad x_1 - x_2 \leq 0.5 \\
& \quad x_1 + x_2 \leq 1.5 \\
& \quad x_1 \in X_1, x_2 \in X_2
\end{align*}
\]

(A.1)

with $X_i = \{x_i \in \mathbb{Z} \mid 0 \leq x_i \leq 1\}, \ i = 1, 2$. Figure A.1(a) depicts its geometry. The only two feasible (and also optimal) points for this problem are $(x_1, x_2) = (0, 0)$ and $(0, 1)$. Relaxing the constraints $x_1 - x_2 \leq 0.5$ and $x_1 + x_2 \leq 1.5$, and introducing the corresponding multipliers $(\lambda_1, \lambda_2) \geq 0$, leads to the dual function

\[
d(\lambda) = \begin{cases} 
-0.5\lambda_1 + 0.5\lambda_2 - 1 & \lambda \in \{1\} \\
-1.5\lambda_1 - 0.5\lambda_2 & \lambda \in \{2\} \\
-0.5\lambda_1 - 1.5\lambda_2 & \lambda \in \{3\} \\
+0.5\lambda_1 - 0.5\lambda_2 - 1 & \lambda \in \{4\},
\end{cases}
\]

where the regions are arranged according to Figure A.1(b). It can be seen that any point on the intersection of regions 1 and 4 attains the largest value for the dual function, and is therefore a solution for the dual problem, i.e.,

\[
\begin{bmatrix}
\lambda_1^* \\
\lambda_2^*
\end{bmatrix} = \begin{bmatrix}
1 \\
1
\end{bmatrix} \cdot \theta, \ \theta \in [0, 0.5].
\]

For any $\theta \in [0, 0.5)$ the set of primal solutions recovered is

\[
\mathcal{X}_1(\lambda^*) = \arg\min_{x_1 \in X_1} \{x_1(\lambda_1^* + \lambda_2^* - 1)\} = \{1\}
\]

\[
\mathcal{X}_2(\lambda^*) = \arg\min_{x_2 \in X_2} \{x_2(-\lambda_1^* + \lambda_2^*)\} = \{0, 1\}.
\]

Thus, at these dual optimizers, the inner solution is non-unique, and for any selection made, e.g. $x_1(\lambda^*) = 1 \in \mathcal{X}_1(\lambda^*)$ and $x_2(\lambda^*) = 0 \in \mathcal{X}_2(\lambda^*)$, the pair $(x_1(\lambda^*), x_2(\lambda^*))$ is infeasible for problem (A.1).
A.2 Primal Recovery from $\lambda^*$ in the Convex Case: Countereample

It is a somewhat spread opinion that primal recovery issues only affect non-convex problems, such as in the discrete optimization case. We provide a simple LP as a counterexample to this.

Consider the problem

$$\begin{align*}
\text{minimize} & \quad -x_1 \\
\text{subject to} & \quad x_1 + x_2 \leq 1.5 \\
& \quad 0 \leq x_1, x_2 \leq 1
\end{align*}$$

whose geometry is depicted in Figure A.2. We dualize $x_1 + x_2 \leq 1.5$ and obtain
$d(\lambda) = \min_{0 \leq x_1, x_2 \leq 1} -x_1 + \lambda(x_1 + x_2 - 1.5)$

$$= \begin{cases} 
-1 - 0.5\lambda & 0 \leq \lambda \leq 1 \\
-1.5\lambda & \lambda > 1.
\end{cases}$$

The unique optimal dual solution is $\lambda^* = 0$, and the set of primal solutions recovered from $\lambda^*$ is

$$\mathcal{X}(\lambda^*) = \left\{ x \in \mathbb{R}^2 \left| \begin{array}{c}
x_1 = 1 \\
0 \leq x_2 \leq 1
\end{array} \right. \right\}$$

Notice how half of the solutions are infeasible for the initial problem. By making one of the optimization variables unbounded, it is even easy to construct counterexamples in which the primal solutions recovered can be arbitrarily far from the feasible set.

This shows that convexity and zero duality gap alone are not sufficient for ensuring primal recovery. For LPs, in some literature this is known as the “uncoordinability of Linear Programming”, see for example [Larsson and Liu, 1997].

It is however worth noting that in the convex case we at least have that $X^* \subseteq \mathcal{X}(\lambda^*)$, where $X^*$ is the set of optimizers to the primal problem [Rockafellar, 1997b, Thm. 28.1]. This does not hold in the linear mixed integer case, unless the duality gap is zero.

### A.3 Analytical Example of Theorem 4.1

With this analytical example we verify that if we apply the contraction procedure of Theorem 4.1, then any primal solution recovered from any dual optimal one is feasible (albeit suboptimal).

Suppose we have

$$\begin{aligned}
\min & \sum_{i=1}^{4} c_i x_i \\
\text{s.t.} & \sum_{i=1}^{4} H_i x_i \leq 11.1 \\
& x_i \in X_i \quad i = 1, \ldots, 4
\end{aligned}$$

(A.2)
with

\[
X_1 = \left\{ x \in \mathbb{Z}_+^2 \left| \begin{array}{cc} 0 & 1 \\ 1 & 1 \end{array} \right| x \leq \left[ \begin{array}{c} 1.2 \\ 2.1 \end{array} \right] \right\}, \quad c_1 = [1, 1] \quad H_1 = [1, 1]
\]

\[
X_2 = \left\{ x \in \mathbb{Z}_+^2 \left| \begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} \right| x \leq \left[ \begin{array}{c} 0.6 \\ 2.1 \end{array} \right] \right\}, \quad c_2 = [-2, 1] \quad H_2 = [5, 1]
\]

\[
X_3 = \left\{ x \in \mathbb{Z}_+^2 \left| \begin{array}{cc} 1 & 0 \\ -0.5 & 1 \end{array} \right| x \leq \left[ \begin{array}{c} 2.2 \\ 1.1 \end{array} \right] \right\}, \quad c_3 = [0.5, -1] \quad H_3 = [1, 1]
\]

\[
X_4 = \left\{ x \in \mathbb{Z}_+^2 \left| \begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right| x \leq \left[ \begin{array}{c} 1.2 \\ 2 \end{array} \right] \right\}, \quad c_4 = [-3, 0.5] \quad H_4 = [1, 1],
\]

see Figure A.3. Relaxing the constraint \(\sum_{i=1}^{4} H_i x_i \leq 11.1\) in this problem leads to the dual function

\[
d(\lambda) = \begin{cases} 
-8 + 0.9\lambda & 0 \leq \lambda \leq 2/5 \\
-4 - 8.9\lambda & 2/5 < \lambda \leq 1 \\
-3 - 9.9\lambda & 1 < \lambda \leq 3 \\
-10.9\lambda & \lambda > 3 
\end{cases}
\]

so that the optimal dual solution is \(\lambda^* = 2/5\), and \(d(\lambda^*) = J^*_D = -7.64\), while the primal optimal objective is \(J^* = -7\) (note the duality gap). The corresponding set of primal solutions recovered from the dual optimizer is thus

\[
X_1(\lambda^*) = \arg \min_{x_1 \in X_1} \{ c_1 x_1 + (\lambda^*)^T H_1 x_1 \} = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}
\]

\[
X_2(\lambda^*) = \left\{ \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \end{bmatrix} \right\}
\]

\[
X_3(\lambda^*) = \left\{ \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \right\}
\]

\[
X_4(\lambda^*) = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right\}.
\]
A Examples and Counterexamples

If one selects $x_2(\lambda^*) = [2, 0]' \in \mathcal{X}_2(\lambda^*)$, then the tuple $(x_1(\lambda^*), \ldots, x_4(\lambda^*))$ is infeasible for problem (A.2). As opposed to the previous (counter-)example in Section A.1, in this case some solutions are feasible, some are not.

We now apply Theorem 4.1. We have one constraint, hence $m = 1$, and we compute $\rho$ as

$$\rho = m \cdot \max_{i \in I} \left( \max_{x_i \in X_i} H_i x_i - \min_{x_i \in X_i} H_i x_i \right) = 1 \cdot \max(2, 10, 4, 3) = 10.$$ 

Thus we substitute $\bar{b} = 11.1 - 10 = 1.1$ in $\mathcal{P}$, and the dual function we obtain is

$$\tilde{d}(\lambda) = \begin{cases} 
-8 + 10.9\lambda & 0 \leq \lambda \leq 2/5 \\
-4 + 0.9\lambda & 2/5 < \lambda \leq 1 \\
-3 - 0.1\lambda & 1 < \lambda \leq 3 \\
-1.1\lambda & \lambda > 3.
\end{cases}$$

The dual optimizer for the contracted problem is $\bar{x}^* = 1$, and the primal solutions recovered are

$$\mathcal{X}_1(\bar{x}^*) = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}, \quad \mathcal{X}_2(\bar{x}^*) = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}, \quad \mathcal{X}_3(\bar{x}^*) = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\}, \quad \mathcal{X}_4(\bar{x}^*) = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\}.$$

Any arbitrary selection of $x(\bar{x}^*) \in \mathcal{X}(\bar{x}^*)$ is feasible for the original problem $\mathcal{P}$. The result stated in Theorem 4.1 is verified.

In Figure A.4 we have depicted the dual functions related to this problem, and the contracted dual function we would obtain if in A.2 we had $b = 11$ instead of $b = 11.1$. Based on this picture, we note two things.

First, we see how contracting the budget $b \to \bar{b}$ tilts the dual function upwards, resulting in dual optimizers that are generally larger. More precisely, we have $\rho \tilde{\bar{x}}^* \geq \rho \tilde{x}^*$, and if $\rho = \alpha \cdot 1$, i.e., the contraction is uniform, then $\|\tilde{\bar{x}}^*\|_1 \geq \|\tilde{x}^*\|_1$. We can see this uplift in “prices” as the additional cost of operating non-convex subsystems distributedly (as opposed to centrally), while insisting on global feasibility.

Second, if we had $b = 11$, then the contracted dual function would attain a “flat region” as depicted on the figure, and the dual optimizer would not be unique any more. We
Figure A.4: Dual functions derived from the original and the contracted problem. The contraction has the effect of tilting the dual function upwards.

would have $\bar{\lambda}^{*}_{11} = \{\bar{\lambda}^{*}_{11} \in \mathbb{R} | 1 \leq \bar{\lambda}^{*}_{11} \leq 3\}$, and the primal solutions recovered would be

\[
\begin{align*}
\mathcal{X}_1(\bar{\lambda}^*) &= \begin{cases} 
0 \\
0 
\end{cases} \\
\mathcal{X}_2(\bar{\lambda}^*) &= \begin{cases} 
0 \\
0 
\end{cases} \\
\mathcal{X}_3(\bar{\lambda}^*) &= \begin{cases} 
\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \text{if } \lambda^* = 1, \\
\begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{otherwise} 
\end{cases} \\
\mathcal{X}_4(\bar{\lambda}^*) &= \begin{cases} 
\begin{bmatrix} 0 \\ 1 \end{bmatrix} & \text{if } \lambda^* = 3, \\
\begin{bmatrix} 1 \\ 0 \end{bmatrix} & \text{otherwise.} 
\end{cases}
\end{align*}
\]

Notice that even though $\bar{\lambda}^*$ is not unique and Assumption 3.1 is violated, any arbitrary selection of $x(\bar{\lambda}^*) \in \mathcal{X}(\bar{\lambda}^*)$ for any $\bar{\lambda}^* \in \bar{\lambda}^*$ would still be feasible for the original problem $\mathcal{P}$. However, in the next section we show that 3.1 is indeed necessary to ensure feasibility.
A.4 Analytical Counterexample to Theorem 4.1 in Absence of Assumption 3.1

In this counterexample we show how Theorem 4.1 may fail in the absence of Assumption 3.1.

Consider again problem (A.2), but now with \( b = 10 \) and the subsystems defined as

\[
X_i = \left\{ x_i \in \mathbb{Z}_+^2 \left| \begin{array}{c} 0 \\ 0 \end{array} \right. \leq x_i \leq \begin{bmatrix} 3.2 \\ 1.4 \end{bmatrix} \right\} \quad c_i = [-1, 1] \quad H_i = [1, 1], \quad i = 1, \ldots, 4.
\]

Notice that since all the subsystems are identical, the problem is highly symmetric. Dualizing the coupling constraint leads to the following dual function

\[
d(\lambda) = \begin{cases} -12 + 2\lambda & 0 \leq \lambda \leq 1 \\ -10\lambda & \lambda > 1, \end{cases}
\]

while when we substitute \( \tilde{b} = b - \rho = 10 - 4 = 6 \), the dual function becomes

\[
\tilde{d}(\lambda) = \begin{cases} -12 + 6\lambda & 0 \leq \lambda \leq 1 \\ -6\lambda & \lambda > 1. \end{cases}
\]

In both cases, the unique optimal dual solution is \( \lambda^* = \tilde{\lambda}^* = 1 \). However, \( \tilde{x}^*_L \) is not unique: for example

\[
(\tilde{x}^*_L)_1 = \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \quad (\tilde{x}^*_L)_2 = \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \quad (\tilde{x}^*_L)_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad (\tilde{x}^*_L)_4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]

and

\[
(\tilde{x}^*_L)_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \quad (\tilde{x}^*_L)_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \quad (\tilde{x}^*_L)_3 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \quad (\tilde{x}^*_L)_4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]

are both valid optimizers of \( \mathcal{P}_L \). Hence, Assumption 3.1 is violated. The primal points that can be recovered from the contracted dual are

\[
\mathcal{X}(\tilde{\lambda}^*) = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 3 \\ 0 \end{bmatrix} \right\}, \quad i = 1, \ldots, 4.
\]

Notice now how the selection \( \chi(\tilde{\lambda}^*) = [3, 0]^T \in \mathcal{X}(\tilde{\lambda}^*) \), for \( i = 1, \ldots, 4 \) is infeasible for the primal problem. This shows that Theorems 3.2 and 4.1 do not hold without Assumption 3.1.
B. Optimal Local Charging Policy for the EVs

The local problem to be solved is

\[
\begin{align*}
\min & \quad P \sum_{k=0}^{N-1} u_i[k] \psi_i[k] \\
\text{subject to} & \quad e_i[k] = E_i^{\text{init}} + \sum_{t=0}^{k}(P_i \Delta T \zeta_i) u_i[k] \\
& \quad e_i \leq E_i^{\text{max}} \\
& \quad e_i[N_i] \geq E_i^{\text{ref}} \\
& \quad u_i \in \{0, 1\},
\end{align*}
\]

in which we have defined the “price profile” seen by EV \( i \) as \( \psi_i \equiv \delta_i + \lambda_1 - \lambda_2 + \lambda_3 \cdot \mathbb{I}(I^c) \).

We furthermore define the minimum and maximum number of charging steps as

\[
C_i^{\text{min}} = \left\lceil \frac{E_i^{\text{ref}} - E_i^{\text{init}}}{P_i \Delta T \zeta_i} \right\rceil, \quad C_i^{\text{max}} = \left\lceil \frac{E_i^{\text{max}} - E_i^{\text{init}}}{P_i \Delta T \zeta_i} \right\rceil.
\]

We see that if \( \psi_i \geq 0 \), then the optimal control strategy is to activate charging for the least possible number of steps \( C_i^{\text{min}} \), and to do so during those times when \( \psi_i \) is lowest.

For a price profile which contains possibly negative values, we can extend this as follows. We sort the entries of \( \psi_i \) in ascending order, storing it in \( \psi_i^j \), and introduce

\[
C_i^j \equiv \max \left\{ k \mid \psi_i^j[k] < 0 \right\},
\]

\[
C_i = \max \left\{ C_i^{\text{min}}, \min \left\{ C_i^0(\psi_i^j), C_i^{\text{max}} \right\} \right\}.
\]

Then,

\[
u_i^*[k] = \begin{cases} 1 & \psi_i[k] \leq \psi_i^j[C_i] \\ 0 & \text{otherwise} \end{cases}.
\]
C. Proofs

C.1 Proof of Fact 3.1

Proof. Due to the linearity of the objective function and the definition of the set $X_i$, it is straightforward to observe that

$$
\min_{x_i \in X_i} (c_i^T + \lambda^T H_i) x_i = \min_{x_i \in \text{conv}(X_i)} (c_i^T + \lambda^T H_i) x_i = \min_{x_i \in \text{vert}(X_i)} (c_i^T + \lambda^T H_i) x_i.
$$

(C.1)

Thus, the desired assertion readily follows from the fact that $X_i$ are non-empty. $\square$

C.2 Proof of Theorem 3.2

Proof. Let us introduce two new LPs that are crucial for our subsequent analysis. First, we denote by $x_i^j$ the $j$-th element of $\text{vert}(X_i)$ for $j \in J_i$ where $J_i = \{1, \ldots, |\text{vert}(X_i)|\}$. In view of (C.1), one can derive an LP version of the program $D$ as

$$
\begin{align*}
\max_{\lambda} & -\lambda^T b + \sum_{i \in I} \min_{j \in J_i} \left( c_i^T x_i^j + \lambda^T H_i x_i^j \right) \\
\text{subject to} & \quad \lambda \geq 0,
\end{align*}
$$

which can then be cast as the LP

$$
\begin{align*}
\max_{\lambda, z, s} & -\lambda^T b + \sum_{i \in I} z_i \\
\text{subject to} & \quad z_i = c_i^T x_i^j + \lambda^T H_i x_i^j - s_i^j, \quad i \in I, j \in J_i \\
& \quad s_i^j \geq 0, \quad i \in I, j \in J_i \\
& \quad \lambda \geq 0,
\end{align*}
$$

(Dlp)

where $s_i^j$ is the slack variable, and $z_i$ corresponds to the inner problem $\min_{j \in J_i} (c_i^T x_i^j + \lambda^T H_i x_i^j)$. The second LP is the dual program of $Dlp$ described as

$$
\begin{align*}
\min_{p_i} & \sum_{i \in I} \sum_{j \in J_i} p_i c_i^T x_i^j \\
\text{subject to} & \quad \sum_{i \in I} \sum_{j \in J_i} p_i H_i x_i^j \leq b \\
& \quad \sum_{j \in J_i} p_i = 1, \quad i \in I \\
& \quad p_i \geq 0, \quad i \in I, j \in J_i.
\end{align*}
$$

(Plp)
where \( p^*_j \in [0,1] \) is the scalar optimization variable associated to the vertex \( x^*_j \). Let us denote by \( p^* \) an optimizer of \( P_{lp} \). Note that \( P_{lp} \) corresponds to an extended LP version of \( P_{LP} \), yet they are not entirely equivalent problems. In particular, each \( p^* \) leads to a unique \( x^*_p \), but the reverse does not hold, i.e., uniqueness of \( x^*_p \) does not imply uniqueness of \( p^* \). We split the proof of the theorem by proving the following steps:

(a) Let \( I_1 \subset I \) be a subset of indices where \( (x^*_p)_i \in \text{vert}(X_i) \) for all \( i \in I_1 \). Then, \( (x^*_p)_i \) is an optimizer of the inner problem, i.e., \( (x^*_p)_i \in \arg\min_{x_i \in X_i} (c_i^T x + \lambda^* H_i x) \) where \( \lambda^* \) is an optimizer of \( D \).

(b) Let \( (\lambda^*, z^*, s^*) \) be an optimal solution of \( D_{ip} \) and \( p^* \) be an optimal solution of \( P_{lp} \) with the corresponding optimizer \( x^*_p \) for \( P \). If the optimal pair \( (p^*, s^*) \) is strictly complementary, then \( (x^*_p)_i = x_i(\lambda^*) \) for all \( i \) in the subset \( I_1 \) as defined in (a).

(c) If \( x^*_p \) is a vertex for the program \( P \), then the subset \( I_1 \) in (a) can be selected such that \(|I_1| \geq |I| - m\).

Before proceeding with the proofs of the above results, let us highlight how the desired assertion, under the unique primal and dual optimizers, follows from these three steps. First, note that if the optimal solution of \( D \) is unique, then \( (\lambda^*, z^*, s^*) \) is the unique solution to \( D_{ip} \): \( \lambda^* \) coincides for \( D \) and \( D_{ip} \) according to [Geoffrion, 1974, p. 89]; \( z^* \) is the optimal objective of the \( i \)-th inner problem, and is thus uniquely determined for fixed \( \lambda \); and finally \( (s^*)_i \) is also uniquely determined by the equality constraints in \( D_{ip} \), in which it is the only variable left undetermined. Therefore, \( s^* \) always belongs to the pair \((p^*, s^*)\) of primal-dual optimizers for which strict complementarity holds; the existence of such a pair is guaranteed in the LP setting [Greenberg, 1994, Thm. 2.1]. Moreover, if \( x^*_p \) is unique, then it is always a vertex. Hence, the requirements of the above results are fulfilled and the theorem assertion is concluded.

**Proof of (a):** Let \( (x^*_p)_i \in \text{vert}(X_i) \). Then, owing to the uniqueness of \( x^*_p \), for any solution \( p^* \) of \( P_{lp} \) we have \( (p^*)_j = 1 \) for the corresponding \( j \in J \). Therefore, by complementary slackness, the dual optimizer has \( (s^*)_j = 0 \), and the step (a) follows by

\[
z^*_j = c^*_j (x^*_p)_j + \lambda^* H_j (x^*_p)_j \leq c^*_j x^*_j + \lambda^* H_j x^*_j, \quad \forall j \in J.
\]

**Proof of (b):** Let \( i \in I_1 \) and, as explained in the proof of (a), \( (p^*)_j = 1 \) for the corresponding \( j \in J \). In light of the equality constraint \( \sum_{j \neq j} p^*_j = 1 \), we have \( (p^*)_j = 0 \) for all \( j \neq j \). The assumed strict complementarity now implies \( (s^*)_j \neq 0 \) for all \( j \neq j \), which leads to a strict inequality in (C.2). Hence, the inner problem \( \min_{x_i \in X_i} (c_i^T x + \lambda^* H_i x) \) has the unique solution \( x_i(\lambda^*) \). Now the desired assertion follows from the step (a).
Proof of (c): Problem $\mathcal{P}_l$ has $m$ inequality constraints ($b \in \mathbb{R}^m$) and $|I|$ equality constraints, plus the positivity constraints on $p_i^j$. We can add slack variables to the complicating constraints thus obtaining a problem with $|I| + m$ equality constraints and positivity constraints on all the optimization variables, which are now the slacks $q \in \mathbb{R}_+^m$ and the variables $p_i^j$. The constraints of $\mathcal{P}_l$ can therefore be rewritten as $H(p^T, q^T)^T = (b^T, 1 \ldots, 1)^T$, $p, q \geq 0$, where the matrix $H$, is defined as

$$H = \begin{bmatrix}
H_{11} x^1_1 & \cdots & \cdots & H_{1|I|} x^1_{|I|} & 0 & \cdots & 0 & 0 \\
1 & \cdots & \cdots & 0 & \cdots & 0 & 0 \\
\vdots & \ddots & \ddots & \vdots & \ddots & \vdots & \vdots \\
0 & \cdots & 0 & \cdots & 0 & 1 & \cdots & 1 \\
\end{bmatrix}_{|I| \times m}, \quad (C.3)
$$

in which we have also defined the submatrices $H_{ii}$, $i \in I$. It is well known (see [Bertsekas, 2009, Prop. 2.1.4 (b)]) that for a problem in this form any feasible point is a vertex if and only if the columns of $H$ corresponding to the non-zero coordinates of the point are linearly independent. This is then true for any optimal vertex. Thus, $\text{supp}(p^*) \leq |I| + m$, as the number of rows of $H$ is $|I| + m$. On the other hand, the constraint $\sum_{i \in I} p_i^j = 1$, $i \in I$ in $\mathcal{P}_l$ forces any feasible solution to have at least one variable $p_i^j$ larger than zero for each $i \in I$, i.e. $\text{supp}(p^*) \geq |I|$. It thus follows that at least $|I| - m$ entries must be set to 1 at any feasible vertex solution, including an optimal one.

\[\square\]

C.3 Proof of Theorem 4.1

Proof. Note that by construction $x(\bar{x}^*) \in \mathcal{X}(\bar{x}^*)$ for all $i \in I$. Then, it only suffices to show $\sum_{i \in I} H_i x_i(\bar{x}^*) \leq b$. By virtue of Theorem 3.2, we know that there exists a subset $I_1 \subseteq I$ such that $|I_1| \geq |I| - m$ and $x(\bar{x}^*) = (\bar{x}^*_{LP})$. Setting $I_2 = I \setminus I_1$, we have

$$\sum_{i \in I} H_i x_i(\bar{x}^*) = \sum_{i \in I_1} H_i x_i(\bar{x}^*) + \sum_{i \in I_2} H_i x_i(\bar{x}^*) = \sum_{i \in I_1} H_i (\bar{x}^*_{LP}) + \sum_{i \in I_2} H_i x_i(\bar{x}^*)$$

$$\leq \sum_{i \in I_1} H_i (\bar{x}^*_{LP}) + \sum_{i \in I_2} (H_i x_i(\bar{x}^*) - H_i (\bar{x}^*_{LP})) \leq b.$$

\[\square\]
C.4 Proof of Theorem 4.2

Proof. Note that

\[ J_P(x(\tilde{x}^*)) - J_P^* = \left[ J_P(x(\tilde{x}^*)) - J_{P_{LP}}^* \right] + \left[ J_{P_{LP}}^* - J_{P_{LP}} \right] + \left[ J_{P_{LP}} - J_P \right], \]

where each term can be bounded as follows:

(i) According to Theorem 3.2, there exists an index set \( I_1 \) with \(|I_1| \geq |I| - m \) such that, for all \( i \in I_1 \), \((\tilde{x}_{1P}^*)_i = x_i(\tilde{x}^*)\). Defining \( I_2 = I \setminus I_1 \), we have

\[ J_P(x(\tilde{x}^*)) - J_P(\tilde{x}_{1P}^*) = \sum_{i \in I_2} \left( c_i^T x_i(\tilde{x}^*) - c_i^T (\tilde{x}_{1P}^*)_i \right) \leq m \cdot \max_{i \in I} \left( \max_{x_i \in X_i} c_i^T x_i - \min_{x_i \in X_i} c_i^T x_i \right) = m \max_{i \in I} \gamma_i. \]

(ii) By virtue of [Nedic and Ozdaglar, 2009, Lemma 1], given the Slater’s point \( \tilde{x} \) we can bound \( \|\lambda^*\|_1 \) by

\[ \|\lambda^*\|_1 \leq \frac{1}{\zeta |I|} \left( \sum_{i \in I} c_i^T \tilde{x}_i - \left( \sum_{i \in I} \min_{x_i \in X_i} \left( c_i^T + \lambda H_i \right) x_i \right) - \lambda^T b \right), \quad \forall \lambda \geq 0. \]

Setting \( \lambda = 0 \) in the above, we arrive at

\[ \|\lambda^*\|_1 \leq \frac{1}{\zeta} \max_{i \in I} \gamma_i, \quad \gamma_i = \max_{x_i \in X_i} c_i^T x_i - \min_{x_i \in X_i} c_i^T x_i. \]

In light of perturbation theory [Boyd and Vandenberghe, 2004a, Sec. 5.6.2], one can bound the term (ii) from above by \((\tilde{x}^*)^T \rho\), where \( \tilde{x}^* \) is the optimizer of the program \( D \) and \( \rho \) is the contraction vector as defined in (4.1). Thus,

\[ J_{P_{LP}}^* - J_{P_{LP}} \leq (\tilde{x}^*)^T \rho \leq \|\lambda^*\|_1 \|\rho\|_\infty \leq \frac{\|\rho\|_\infty}{\zeta} \max_{i \in I} \gamma_i. \]

(iii) By definition, \( P_{LP} \) is a relaxed version of \( P \). Hence \( J_{P_{LP}}^* - J_P^* \leq 0. \)
**C.5 Proof of Theorem 4.3**

Proof. For a given $x(\lambda^*) \in X(\lambda^*)$, let us introduce $\tilde{I} = \{i \in I \mid (x^*_i)_{LP} \neq x_i(\lambda^*)\}$. For the $k$-th complicating constraint we then have

$$\sum_{i \in I} H^k_i x_i(\lambda^*) = \sum_{i \in I \setminus \tilde{I}} H^k_i (x^*_i)_{LP} + \sum_{i \in \tilde{I}} H^k_i x_i(\lambda^*) \leq b + \sum_{i \in \tilde{I}} H^k_i (x_i(\lambda^*) - (x^*_i)_{LP}) = b + \sum_{i \in [I \setminus I_k]} H^k_i (x_i(\lambda^*) - (x^*_i)_{LP}) \leq b + |I \setminus I_k| \max_{i \in [I \setminus I_k]} \left( \max_{x_i \in \mathcal{X}_i} H^k_i x_i - \min_{x_i \in \mathcal{X}_i} H^k_i x_i \right).$$

In order to get a bound on $|I \cap I_k|$, we resort again to the program $P_{LP}$. We know that, under Assumption 3.1, $x_i(\lambda^*) = (x^*_i)_{LP}$ if and only if $(x^*_i)_{LP} \notin \text{vertices}(X_i)$, as shown in Appendix C.2. Thus, if $i \in I$ there are at least two $j \in J_i$ such that $(\rho^*)_j > 0$ in the corresponding program $P_{LP}$. And for every $i \in I$, there is always at least one $j \in J_i$ such that $(\rho^*)_j > 0$. Thus

$$|\text{supp}((\rho^*)_{i \in [I_k]})| \geq |I_k \setminus I| + 2|I \cap I_k| = |I_k| + |I \cap I_k|.$$

On the other hand, in view of [Bertsekas, 2009, Prop. 2.1.4 (b)], and as discussed in Appendix C.2, the columns within the matrix $\mathbb{H}$ (defined in Equation (C.3)) corresponding to non-zero $(\rho^*)_j$ coordinates must be linearly independent. Hence $|\text{supp}(\rho^*)| \leq \text{rank}(\mathbb{H})$ and in particular

$$|\text{supp}((\rho^*)_{i \in [I_k]})| \leq \text{rank}(\mathbb{H}_{i \in [I_k]}).$$

Finally, from the structure of $\mathbb{H}$ defined in Equation (C.3), it is clear that

$$\text{rank}(\mathbb{H}_{i \in [I_k]}) \leq \text{rank}(\mathbb{H}_{i \in [I_k]}) + |I_k|.$$

Combining the above inequalities immediately leads to

$$\text{rank}(\mathbb{H}_{i \in [I_k]}) \geq |I \cap I_k|,$$

as desired. \qed

**C.6 Proof of Proposition 4.1**

The objective is to establish a connection from the sensitivity of the large scale, but structured, optimization program $P_{LP}$ to a reduced version in which only $m$ subsystems appear. To this end, we first start with some preparatory lemmas.
Lemma C.1. Let $J : \mathbb{R}_+ \to \mathbb{R}$ be a convex function. Suppose there exist a constant $L$ and a sequence $\{\epsilon_n\}_{n \in \mathbb{N}}$ such that $\epsilon_n \to 0$ as $n$ goes to infinity and $J(0) - J(\epsilon_n) \leq L\epsilon_n$ for all $n \in \mathbb{N}$. Then, $J(0) - J(\epsilon) \leq L\epsilon$ for all $\epsilon \in \mathbb{R}_+$.

Proof. For the sake of contradiction, suppose there exists an $\epsilon$ such that $J(0) - J(\epsilon) > L\epsilon$. Let $n$ be large enough so that $\epsilon_n \in (0, \epsilon)$ and $\alpha \equiv \frac{\epsilon_n}{\epsilon}$. In light of convexity of $J$, we have

$$J(\epsilon_n) \leq (1 - \alpha)J(0) + \alpha J(\epsilon) < (1 - \alpha)J(0) + \alpha (J(0) - L\epsilon) = J(0) - L\epsilon_n,$$

which is obviously in contradiction with our assumption. $\square$

Lemma C.2. Consider the parametrized LP

$$\begin{cases}
\min \quad cx \\
\text{s.t.} \quad Ax \leq b + \epsilon \mathbf{1},
\end{cases} \tag{C.4}$$

where $\epsilon \in \mathbb{R}_+$ is the parameter and $\mathbf{1} \equiv [1, \ldots, 1]^\top \in \mathbb{R}^m$. Suppose the program admits a vertex optimizer whose objective value is denoted by $J(\epsilon)$. Then, there exists a constant independent of the resource vector $b$, denoted by $L(A, c)$, such that

$$0 \leq J(0) - J(\epsilon) \leq L(A, c)\epsilon, \quad \forall \epsilon \in \mathbb{R}_+.$$

Proof. We only need to prove the right-hand side of the inequality as the left-hand side trivially holds since the parameter $\epsilon$ is non-negative and only relaxes the constraint. Let $x^*(\epsilon)$ be a vertex optimizer for (C.4). By virtue of [Bertsekas, 2009, Prop. 2.1.4 (a)], given a fixed $\epsilon$, we know that there exists a collection of $m$ linearly independent rows of the matrix $A$, denoted by the invertible submatrix $[A](\epsilon)$, such that $[A](\epsilon)x^*(\epsilon) = b + \epsilon \mathbf{1}$. Note that the number of submatrices of matrix $A$ is, of course, finite. Therefore, one can always pick a sequence $\{\epsilon_n\}_{n \in \mathbb{N}}$ such that $\epsilon_n \to 0$ as $n$ goes to infinity and the corresponding submatrix $[A](\epsilon_n)$ is constant; let us denote this submatrix by $[A]$. We thus have

$$J(0) - J(\epsilon_n) = cx^*(0) - cx^*(\epsilon_n) = -c[A]^{-1}\mathbf{1}\epsilon_n \leq L(c, A)\epsilon_n,$$

where the constant can be, for example, $L(c, A) \equiv m\|c\|_2\|A\|^{-1}_2$. Note that, by construction, the submatrix $[A]$ is invertible and the norm $\|A^{-1}\|$ is bounded. The desired assertion now follows from the convexity of the perturbation mapping $\epsilon \mapsto J(\epsilon)$ [Boyd and Vandenberghe, 2004a, Sec. 5.6.2] and Lemma C.1. $\square$
Theorem 4.1. Given the partition \( I = I_1 \cup I_2 \), we introduce a reduced version of \( \mathcal{P}_{LP}(\epsilon) \) associated with the index set \( I_2 \) as follows:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in I_2} c_i^T x_i \\
\text{subject to} & \quad \sum_{i \in I_2} H_i x_i \leq b - \sum_{i \in I_1} H_i (x_{LP}^*)_i + \epsilon 1 \\
& \quad x_i \in \text{conv}(X_i) \quad i \in I_2,
\end{align*}
\]

where \( x_{LP}^* \) is an optimizer of the program \( \mathcal{P}_{LP} \). We denote the optimal value of \( \mathcal{R}_{I_2}(\epsilon) \) by \( J_{\mathcal{R}_{I_2}}(\epsilon) \). Let us highlight that for any partition of the index set \( I = I_1 \cup I_2 \) the program \( \mathcal{R}_{I_2}(\epsilon) \) is always feasible as \( (x_{LP}^*)_i \) trivially satisfies the constraints for any \( \epsilon \in \mathbb{R}_+ \). As a first step in the proof, we show that there exist an index subset \( I_2 \) and a sequence of \( \{\epsilon_n\}_{n \in \mathbb{N}} \) such that \( |I_2| \leq m \) and the optimal values \( J_{\mathcal{P}_{LP}}(\epsilon_n) \) and \( J_{\mathcal{R}_{I_2}}(\epsilon_n) \) have the same sensitivity in terms of the parameter \( \epsilon \).

Let \( x_{LP}^*(\epsilon) \) be a vertex optimizer of the program \( \mathcal{P}_{LP}(\epsilon) \); the existence of such a vertex is always ensured since the feasible set of \( \mathcal{P}_{LP}(\epsilon) \) is a compact polytope. In light of part (c) in the proof of Theorem 3.2, we know that for each \( x_{LP}^*(\epsilon) \) there exists a partition \( I = I_1(\epsilon) \cup I_2(\epsilon) \) where \( |I_2(\epsilon)| \leq m \) and \( (x_{LP}^*(\epsilon))_i \in \text{vert}(X_i) \) for all \( i \in I_1(\epsilon) \). Due to the fact that the number of the subsets of \( I \) as well as the set \( \text{vert}(X_i) \) is finite, then there exists a partition \( I = I_1 \cup I_2 \) and a sequence of \( \{\epsilon_n\}_{n \in \mathbb{N}} \) such that \( |I_2| \leq m \) and \( (x_{LP}^*(\epsilon_n))_i \) are constants for \( i \in I_1 \). By compactness we can, without loss of generality, assume that this sequence is convergent. It is a well-known result in the context of perturbation theory that the mapping \( \epsilon \mapsto J_{\mathcal{P}_{LP}}(\epsilon) \) is convex on \([0, \infty)\), and in particular continuous [Rockafellar, 1997a, Sec. 28]. Hence, one can infer that \( (x_{LP}^*(\epsilon_n))_i \) converges to an optimizer of \( \mathcal{P}_{LP} \), which consequently implies \( (x_{LP}^*(\epsilon_n))_i = (x_{LP}^*_i) \), for all \( i \in I_1 \). Therefore, by construction of the auxiliary program \( \mathcal{R}_{I_2}(\epsilon) \) we can deduce

\[
J_{\mathcal{P}_{LP}}(0) - J_{\mathcal{P}_{LP}}(\epsilon_n) = J_{\mathcal{R}_{I_2}}(0) - J_{\mathcal{R}_{I_2}}(\epsilon_n), \quad \forall n \in \mathbb{N}.
\]

Now, in view of Lemma C.2, we know that the right-hand side of the above equality is non-negative and can be upper bounded by a constant only depending on the data of the subsystems indexed in \( I_2 \), i.e., \( (\mathcal{D}i)_{i \in I_2} \). Let us denote this constant by \( L(I_2) \). Then, we have

\[
0 \leq J_{\mathcal{P}_{LP}}(0) - J_{\mathcal{P}_{LP}}(\epsilon_n) \leq L(I_2)\epsilon_n, \quad \forall n \in \mathbb{N},
\]

that by virtue of Lemma C.1 leads to the desired assertion. \( \square \)
C.7 Proof of Theorem 6.1

Our theorem relies on the following crucial result, known in the literature as the Shapley–Folkman–Starr Theorem.

**Theorem C.1** (Shapley–Folkman–Starr). Let \( S_i \subseteq \mathbb{R}^{m+1}, i \in I, \) be nonempty sets with \(|I| > m+1\), and let \( S = S_1 + \cdots + S_{|I|} \). Then every vector \( s \in \text{conv}(S) \) can be represented as \( s = s_1 + \cdots + s_{|I|} \), where \( s_i \in \text{conv}(S_i) \) for all \( i \in I \), and \( s_i \notin S_i \) for at most \( m+1 \) indices \( i \).

**Proof.** See [Bertsekas, 2009, Prop. 5.7.1]. □

**Theorem 6.1.** Let \( F(P_{LP}) \) be the feasible set of the program \( P_{LP} \), i.e.,

\[
F(P_{LP}) = \left\{ x = \begin{bmatrix} x_1 \\ \vdots \\ x_{|I|} \end{bmatrix} \mid \sum_{i \in I} H_i x_i \leq b, \ x_i \in \text{conv}(X_i) \right\}.
\]

Suppose that \( \bar{x} \doteq [\bar{x}_1^T, \cdots, \bar{x}_{|I|}^T]^T \) is a vertex of \( F(P_{LP}) \). Then there exists a vector \( \bar{c} \doteq [\bar{c}_1^T, \cdots, \bar{c}_{|I|}^T]^T \) such that \( \bar{x} \) is the unique optimizer of

\[
\bar{x} = \arg\min_{x \in F(P_{LP})} \bar{c}^T x.
\]

Consider now the set \( S \doteq S_1 + \cdots + S_{|I|} \) where

\[
S_i = \left\{ \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \in \mathbb{R}^{m+1} \mid s_1 = H_i x_i, \ s_2 = \bar{c}_i^T x_i, \ \text{for some } x_i \in X_i \right\}.
\]

Since \( \text{conv}(H \cdot X) = H \cdot \text{conv}(X) \) (where \( H \) is a matrix, \( X \) is a set – possibly non-convex – and the product is intended to be the multiplication of each vector in \( X \) by \( H \)), we have that

\[
\text{conv}(S_i) = \left\{ \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \in \mathbb{R}^{m+1} \mid s_1 = H_i x_i, \ s_2 = \bar{c}_i^T x_i, \ \text{for some } x_i \in \text{conv}(X_i) \right\}.
\]

Then, in view of Theorem C.1, for every \( x = [x_1^T, \cdots, x_{|I|}^T]^T, \ x_i \in \text{conv}(X_i) \), there exist \( I_1 \subseteq I, |I_1| \geq m+1, \) and a representation \( y = [y_1^T, \cdots, y_{|I|}^T]^T \) such that \( y_i \in \text{conv}(X_i) \) for all \( i \in I \), \( y_i \in X_i \) for all \( i \in I_1 \) and

\[
\sum_{i \in I} \tilde{H}_i x_i = \sum_{i \in I} \tilde{H}_i y_i, \quad \sum_{i \in I} \tilde{c}_i^T x_i = \sum_{i \in I} \tilde{c}_i^T y_i.
\]

This implies that the representation \( y \) of any \( x \in F(P_{LP}) \) also belongs to \( F(P_{LP}) \) and attains the same objective value. But since \( \bar{x} \) is the unique minimizer, it must coincide with its representation \( \bar{y} \), which concludes the proof. □
C.8 Proof of Theorem 6.2

We first need the following Lemma.

**Lemma C.3** (unique accumulation point implies convergence). Let \( \{a_n\}_{n \in \mathbb{N}} \) be a sequence in the compact set \( A \), and \( \mathcal{A} \) the set of all its accumulation points. If \( \mathcal{A} \) is singleton, then the sequence \( \{a_n\}_{n \in \mathbb{N}} \) converges.

**Proof.** Since \( A \) is compact, any subsequence has an accumulation point which, by definition, has to be in \( \mathcal{A} \). Then, since \( \mathcal{A} \) is a singleton, i.e., it contains only one point which we denote by \( a \), we can deduce that any subsequence converges to the fixed point \( a \), which leads to the desired assertion. \( \square \)

**Proof of Theorem 6.2.** We first note that, according to the Fundamental Theorem of Linear Programming [Bertsekas, 2009, Prop. 2.4.2], optimizers to the inner problem in the dual of \( P_{LP} \), i.e., optimizers of

\[
\min_{x_i \in \text{conv}(X_i)} (c_i^T + \lambda^T H_i)x_i,
\]

are attained at some vertex of \( X_i \). Thus, an optimizer to the inner problem in the dual of \( P \) is a valid optimizer for (C.5), i.e.,

\[
x_i(\lambda) \in \arg \min_{x_i \in X_i} (c_i^T + \lambda^T H_i)x_i.
\]

According to [Anstreicher and Wolsey, 2009, Thm. 6], it then follows that the sequence \( \{x_i^k\} \) accumulates at the optimizers of \( P_{LP} \). By uniqueness of \( x_i^* \) the accumulation point is unique, and since the sequence \( \{x_i^k\} \) is in a compact set, according to Lemma C.3, it converges, i.e., \( x_i^k \rightarrow x_i^* \).

\( \square \)

C.9 Proof of Theorem 6.3

**Proof.** The recovered solution \( \mathbf{x}^* \) satisfies \( \mathbf{x}^* \in X_i \) (including the the integrality constraints) by construction. In order to prove its feasibility, we only need to show that it
satisfies the coupling constraints:

\[
\sum_{i \in I} H_i \hat{x}_i = \sum_{i \in I_1} H_i \hat{x}_i + \sum_{i \in I_2} H_i \hat{x}_i = \sum_{i \in I_1} H_i (\hat{x}^*_{i,LP}) + \sum_{i \in I_2} H_i \hat{x}_i \leq b, \tag{a}
\]

in which we have used that, for \(i \in I_1\), \(\hat{x}_i = (\hat{x}^*_{i,LP})_i\). The term (a) is determined in the algorithm when a solution to \(P_{\text{FIX}}\) is computed. According to Assumption 6.2, this problem is always feasible since it allows the feasible solution \((\hat{x}^*_{i,LP})_{i \in I_2} = 0\). Hence, the last inequality follows.

For the performance, we first note that

\[
J_P(\hat{x}^*) - J_P^* = \left[ J_P(\hat{x}^*) - J^*_{P_{LP}} \right] + \left[ J^*_{P_{LP}} - J_P^* \right], \tag{C.6}
\]

and then bound each term as follows:

(i) by construction, \(x^*_{i,LP}\) and \(\hat{x}^*\) coincide on \(I_1\), which, according to Theorem 6.2, has a cardinality of at least \(|I| - m\). It thus follows that

\[
J_P(\hat{x}^*) - J^*_{P_{LP}} = \sum_{i \in I_1} c_i^T (\hat{x}_i^* - (\hat{x}^*_{i,LP})_i) + \sum_{i \in I_2} c_i^T (\hat{x}_i^* - (\hat{x}^*_{i,LP})_i) \\
\leq m \cdot \max_{i \in I} \max_{x_i \in X_i} c_i^T x_i,
\]

where the last inequality follows again by setting \((\hat{x}^*_{i,LP})_{i \in I_2} = 0\).

(ii) Problem \(P_{LP}\) is a relaxation of \(P\), such that \(J^*_{P_{LP}} - J_P^* \leq 0\).

C.10 Proof of Theorem 9.1

**Proof.** The first step is to replace the uncertain constraints by a (possibly infinite) number of constraints, one for each realization of the uncertain variable \(w\). The robust counterpart to the uncertain optimization problem (UIP) is then

\[
\min_{x} c^T x \\
\text{subject to} \quad Ax + Dw \leq b \quad \forall w \in W(x) \\
x \in \{0,1\}^n.
\]

\[\tag{C.6}\]
Now, we can eliminate the universal quantifier as follows

\[ Ax + Dw \leq b \quad \forall w \in W(x) \quad \Leftrightarrow \quad Ax + \sup_{w \in W(x)} Dw \leq b. \tag{C.7} \]

Then, we consider Equation (C.7) row-wise and for each row

\[ a_i \cdot x + \sup_{w \in W(x)} d_i \cdot w \leq b_i \quad \forall i \]

where \( a_i \) and \( d_i \) are, respectively, the \( i \)-th row of the matrices \( A \) and \( D \), we examine \( \sup_{w \in W(x)} d_i \cdot w \) and obtain

\[
\sup_{w \in W(x)} d_i \cdot w = (1) \sigma_{W(x)}(d_i) = (2) \sigma_{\oplus_{i=1}^n [k_i] \cdot W_i}(d_i) = (3) \sum_{i=1}^k \sigma_{x_i \cdot W_i}(d_i)
\]

\[
= (4) \sum_{k=1}^n x[k] \cdot \sigma_{W_i}(d_i) = (5) \sum_{k=1}^n x[k] \cdot \sigma_{\text{conv}(W_i)}(d_i),
\]

where

1. (1) is the definition of support function \( \sigma \),
2. (2) follows from the structure imposed on \( W(x) \) in (9.2),
3. (3) follows from the property of support functions that \( \sigma_{X \oplus Y} = \sigma_X + \sigma_Y \), see [Boyd and Vandenberghe, 2004b, Ex. 3.35, p.120],
4. (4) follows since \( \sigma \) is a positively homogeneous function, i.e., \( \sigma_Y(\alpha x) = \sigma_{\alpha Y}(x) = \alpha \sigma_Y(x) \) for \( \alpha \in \mathbb{R}_+ \), see [Rockafellar, 1997b, Thm. 13.2],
5. (5) follows since \( \sigma_Y = \sigma_{\text{conv}(Y)} \) [Boyd and Vandenberghe, 2004b, Ex. 3.35, p.120].

This immediately leads to the desired optimization problem \( (RC - UIP) \).

\[ \square \]

### C.11 Proof of Theorem 9.2

**Proof.** The proof is the same as for Theorem 9.1 up to item (3); neither integrality of \( x \), nor its sign, are used up to that point. The reason why we need to assume the absolute value in the uncertainty set structure is that then we can proceed with step (4) (the multiplicative constant of the set must be positive). We then obtain

\[
\sum_{k=1}^n \sigma_{[k] \cdot W_i}(d_i) = \sum_{k=1}^n |x[k]| \cdot \sigma_{W_i}(d_i) = \begin{cases} 
\sum_{k=1}^n y[k] \cdot \sigma_{W_i}(d_i) & -y \leq x \leq y \\
-y & y \geq 0
\end{cases}
\]

which leads to the desired optimization problem. \[ \square \]


Curriculum Vitae

Robin Vujanic
born 31st July, 1984 in Switzerland.

2009-2013 **Ph.D. in Electrical Engineering**
Automatic Control Laboratory
ETH Zurich, Switzerland

2006-2009 **M.Sc. in Mechanical Engineering, with distinction**
Major in Control and Micro-Nano Systems
ETH Zurich, Switzerland

2003-2006 **B.Sc. in Mechanical Engineering**
ETH Zurich, Switzerland

1999-2003 **High-School Matura**
Liceo Cantonale di Lugano 1, Switzerland